



Universitat Autònoma
de Barcelona

GISCMO, MÒDUL DE MODELATGE D'OBJECTES

Memòria del projecte
d'Enginyeria Tècnica en
Informàtica de Sistemes

realitzat per

Marc Àngel Rodríguez

i dirigit per

Rafael Cortés Fité

Escola d'Enginyeria

Sabadell, *Juny* de 2010

El sotasignat, Rafael Cortés i Fité,
professor de l'Escola d'Enginyeria de la UAB,

CERTIFICA:

Que el treball al que correspon la present
memòria
ha estat realitzat sota la seva direcció
per en ***Marc Àngel Rodríguez***

I per a que consti firma la present.
Sabadell, ***Juny*** de ***2010***

Signat: Rafael Cortés i Fité

Resum:

El projecte es basa en la creació del mòdul de modelatge d'objecte per a GISCMO. GISCMO serà una aplicació web, es vol crear una eina software per donar suport al disseny d'entorns de control industrial SCADA.

El mòdul de modelatge d'objectes consisteix en crear una eina en entorn web capaç de poder crear i modelar objectes, que s'utilitzarà per fer el disseny de projectes de control industrial.

Per poder modelar els objectes, el mòdul de modelatge d'objectes ens permetrà crear un objecte des de 0, associant-li unes caracteritzes, una imatge determinada i un seguit de dades tècniques, a més de senyals digitals, com poden ser: alarmes o senyals de controls.

Tot el projecte es crearà en un entorn operatiu web, amb un sistema de finestres que ens facilitarà la feina, a més els administradors tindran una part per la gestió d'usuaris.

Amb el mòdul de modelatge d'objectes es pretén assolir les necessitats que requereix el modelatge d'objectes per al disseny de control industrial, i formar part de GISCMO.

Taula de continguts

1	Introducció.....	8
1.1	Resum de continguts	8
1.2	Motivacions personals	10
1.3	Estat de l'art	10
1.4	Aproximació històrica a la tecnologia.....	18
1.5	Objectius previstos	23
1.6	Fonaments teòrics	24
1.6.1	Linux:	25
1.6.2	Apache:.....	26
1.6.3	MySQL:.....	27
1.6.4	GISCMO:	27
1.6.5	SCADA:.....	27
1.7	Estructura de la memòria.....	28
2	Estudi de viabilitat.....	29
2.1	Tipologia i paraules clau.	29
2.2	Objecte.....	29
2.2.1	Situació actual.....	29
2.2.2	Perfil d'usuari del projecte.....	30
2.3	Objectius.....	32
2.4	Descripció del sistema	33
2.4.1	Lògica del sistema	33
2.5	Recursos.....	33
2.5.1	Recursos humans:	33
2.5.2	Recursos hardware.....	34
2.5.3	Recursos software.....	34
2.5.4	Recursos amb costs	35

2.6	Planificació del projecte	35
2.6.1	Planificació inicial	35
2.6.2	Descripció de les tasques	36
2.6.3	Diagrama de Gantt.....	37
2.7	Anàlisi de costos	38
2.7.1	Cost de la càrrega de treball o recursos humans.....	38
2.7.2	Cost dels recursos de hardware.....	38
2.7.3	Cost dels recursos de software.....	38
2.7.4	Avaluació de riscos	39
2.8	Conclusions	39
3	Anàlisi del projecte	40
3.1	Introducció.....	40
3.2	Requeriments funcionals.....	40
3.2.1	Diagrames de casos d'ús	41
3.2.2	Diagrames de seqüència	44
3.3	Requeriments no funcionals.....	50
3.3.1	Restriccions de disseny.....	50
3.3.2	Objectius de disseny.....	51
4	Disseny.....	53
4.1	Introducció.....	53
4.2	El model relacional	54
4.2.1	Introducció.....	54
4.2.2	Descripció	54
4.2.3	Esquema.....	55
4.2.4	Instàncies.....	55
4.3	Base de dades relacional	56
4.4	Disseny de la base de dades.....	57

4.5	Descripció de les taules	57
4.5.1	Usuaris	57
4.5.2	Objectes	58
4.5.3	Categoria	59
4.5.4	Característiques	60
4.5.5	Imatges	61
4.5.6	Diccionari	61
4.5.7	Senyals diccionari	61
4.5.8	Senyals	62
4.5.9	Diagrama model de dades	63
4.6	Interfícies	64
4.6.1	Pàgina de login	64
4.6.2	Interfície administrador	65
4.6.3	Gestió d'usuaris	66
4.6.4	Interfície usuari	69
4.6.5	Interfície mòdul modelatge d'objectes GISCMO	69
4.6.6	Finestres menú lateral	74
4.6.7	Senyals	79
4.6.8	Característiques	83
5	Implementació	87
5.1	Introducció	87
5.2	HTML i CSS	87
5.3	Motor d'aplicació	88
5.3.1	Arbre	88
5.3.2	Insertar equip	89
5.3.3	Creació d'una matrícula universal	89
5.3.4	Eliminar usuari	90

5.3.5	Insertar imatge	90
6	Proves.....	93
6.1	Introducció.....	93
6.2	Proves de compatibilitat.....	93
6.3	Proves de seguretat	94
6.4	Proves d'integració del servei	94
7	Conclusions i resultats	95
7.1	Consecució d'objectius	95
7.2	Desviacions.....	95
7.3	Línies d'ampliació	96
7.4	Valoració personal de l'experiència.....	96
8	Bibliografia.....	98
8.1	Enllaços de VMWare	98
8.2	MySQL.....	98
8.3	PHP.....	99
8.4	Javascript	100
8.5	CentOS.....	100
8.6	Programa eclipse.....	100
8.7	HTML.....	101
8.8	SCADA.....	101

1 Introducció

1.1 Resum de continguts

Volem fer un projecte totalment gratuït, utilitzant eines de lliure distribució.

Que és GISCMO?

GISMO pretén ser una eina software per donar suport al disseny d'entorns de control industrial SCADA. Actualment no es pot trobar, en el mercat, un software GNU que sigui capaç de fer el modelatge de sistemes de control, creació de codi automatitzat per diferents sistemes de control, etc. GISCMO serà un projecte Open Source que aportarà una solució a aquest tipus de models.

A més a més, s'utilitzarà un entorn web dinàmic que serà modificable per qualsevol programador a conveni de les seves necessitats.

El projecte GISCMO consta de diversos mòduls:

- Modelatge de hardware
- Modelatge d'objectes
- Gestió i creació de diccionaris de senyals
- Gestió de projectes
- Mòdul de compilació per SCADA o PLC

El nostre projecte consta del desenvolupament d'un dels mòduls de GISCMO, concretament el mòdul de modelatge d'objectes.

El mòdul de modelatge d'objectes, servirà per modelar els elements que utilitzarem posteriorment per fer el disseny dels nostres projectes de control industrial.

Des de el punt de vista de la plataforma GISCMO un objecte es una sèrie d'informació, que especifiquen tan dades tècniques com comercials d'un equip. Aquests equips han de esser entesos com model de elements del mon real com

poden ser motors, vàlvules, llums automàtics, etc. Aquest elements s'agruparan en llibreries que seran d'ús i consulta pública.

Què vol dir, poder modelar objectes? S'ha de poder dissenyar un objecte des de 0, això implicarà poder crear una estructura jeràrquica amb un nombre indeterminat de nivells, que en conjunt modelin un element del món real. S'ha de poder associar una imatge que caracteritzi cada element de la jerarquia. A més, se li han de poder donar uns identificadors únics, una sèrie de característiques i uns senyals les quals estaran definides anteriorment, dins els corresponents diccionaris establerts per el mòdul de gestió de diccionaris, també tenim la possibilitat de crear carpetes per poder organitzar el nostre treball.

Tots els objectes definits mitjança'ns el nostre aplicatiu, estaran gestionats per personal tècnic qualificat que serà l'encarregat i responsable de la creació, classificació i publicació de les llibreries de treball.

1.2 Motivacions personals

Tenir la oportunitat de col·laborar en un projecte gran. Una vegada finalitzat el projecte GISCMO, pot ser de gran utilitat per qualsevol tipus d'empresa implicada en el disseny, control y ús d'un sistema SCADA. Tenir la possibilitat d'aprendre al màxim, entre una gran multitud d'eines que podem utilitzar per aconseguir un bon projecte, des de el codi de programació utilitzant PHP, fins a un servidor virtualitzat.

També entra l'interès per poder participar un projecte que pot evolucionar contínuament, en el que es poden fer modificacions i millorà el servei a comunitat d'usuaris. No és un projecte que tingui un començament i un final, ni un projecte que es pugui queda obsolet en el temps per conseqüència d'una antiga programació.

1.3 Estat de l'art

En el mercat hi ha gran quantitat de software orientat a la programació de PLC's y SCADAS. Per una altra banda, hi ha molts programes de gestió de projectes i control de versions, orientats a plaços d'execució, control de temps i traçabilitat del software. La gran part d'aquest software esta dissenyat per a cobrir arees molt concretes de les diferents etapes de un projecte , però, al tractar-se d'eines generalistes no poden cobrir les necessitats específiques del mon del control industrial.

Els projectes de control industrial han sofert al llarg del temps d'una lacra molt greu derivada de la interdisciplinarietat. La gran quantitat de personal que intervé amb perfils tècnics totalment diferents, com enginyers industrials, químics, informàtics, electricistes, etc., crea una serie problemàtica difícilment controlable. Aquests projectes son molt segmentats en funció de cada fase de procés d'evolució, i per això, cada usuari opta per utilitzar les eines que millor coneixen. En molts casos, aquestes eines de software s'han desenvolupat

enfocades sobre una necessitat en concret, o inclòs, per a donar suport a un hardware determinat.

Aquesta estratificació dels treballs crea veritables abismes, tant de coneixement com de col·laboració entre els diferents participants del projecte.

Els diferents aspectes involucrats en un projecte de control industrial, ha esdevingut a la creació d'eines molt especialitzades, sense vocació de compartir la informació entre elles. A la majoria dels casos les empreses que han invertit els seus esforços i treball en el desenvolupament d'aquestes eines, han establert alts preus i peatges en forma d'estàndards propietaris amb la intenció de mantenir captiu el mercat.

Com a conseqüència d'aquestes actituds de bloqueig i opacitat dels diferents fabricants, s'han reproduït una quantitat inimaginable d'eines, protocols, productes, etc. Sense cap tipus possible d'integració mútua, i buscant imposar-se en el mercat creant estàndards.

En aquesta línia, s'ha de nomenar alguns dels fabricants més importants com SIEMENS AUTOMATION. Aquest, a través d'un titànic esforç en I+D+I, ha establert els seus productes com a estàndard a la gran part de la indústria química. SIEMENS AUTOMATION disposa de donar servei a qualsevol necessitat del mercat, però al preu de ésser un client captiu. A l'avantguarda dels productes dels productes de SIEMENS AUTOMATION per als entorns de control industrial, estan els autòmats de la família S7 en les seves diverses gammes S7-200, S7-300 y S7-400. Aquests productes han demostrat la seva eficiència en gairebé tots els processos industrials des de la química fins al món de l'automoció.



Autòmat SIEMENS AUTOMATION S7-300 amb preferència descentralitzada ET200

Un altre gran fabricant de productes d'automatització es Schneider electric, que és potser el més proper competidor de SIEMENS en el món de l'automatització industrial.

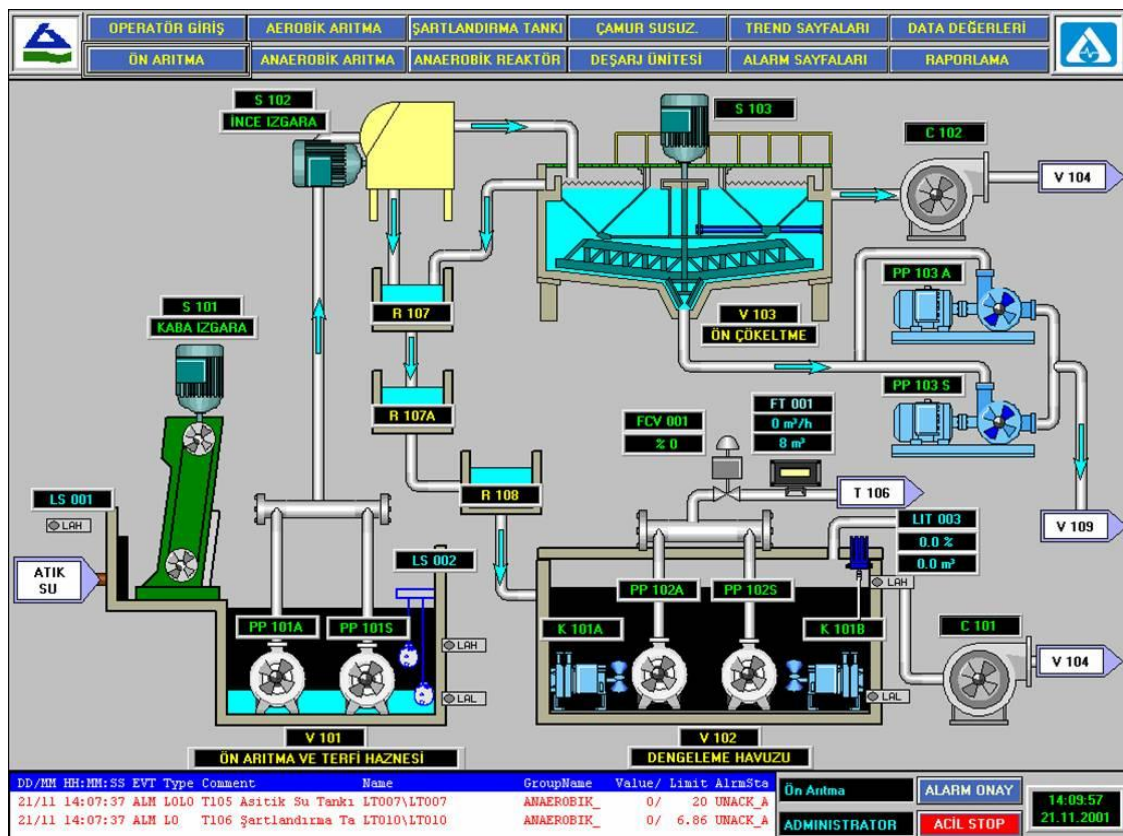
Encara que alguns productes han superat en potència la capacitat d'emmagatzematge als SIEMENS, aquesta marca segueix sent un referent. Alguns dels productes estrella d'Schneider electric estan superant als SIEMENS degut a l'estratègia de mercat que ha impulsat a Schneider a introduir un gran grau de connectivitat estàndard, com el TCP/IP, i el llenguatge de programació C en el entorn de PLC's industrials. Aquesta decisió dels PLC's Schneider està revolucionant el món del control industrial desde fa uns anys.

Un dels PLC's que ha marcat el desplegament de les tecnologies de comunicació en el món del control industrial és el famós Momentum M1 de Schneider, semblant al Pascal que va trencar els esquemes de la majoria dels programadors de PLC acostumats a treballar amb programació ladder logic.

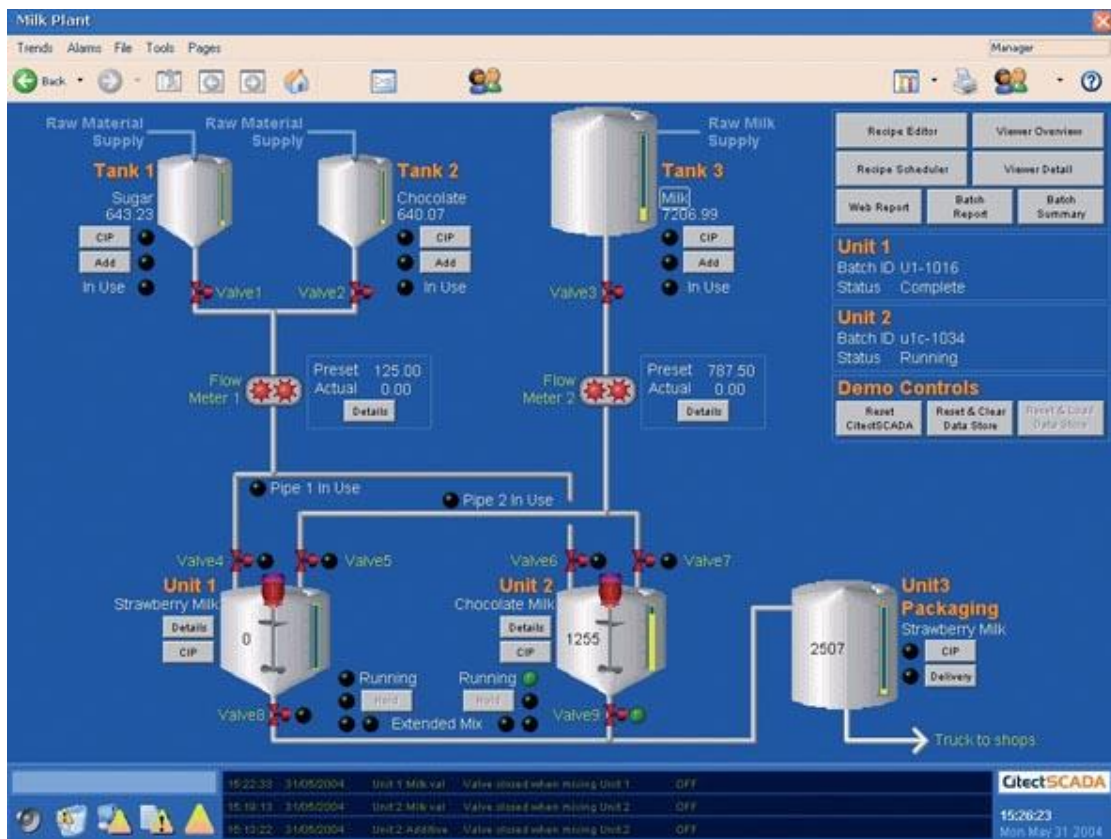


PLC Momentum M1

Poc a poc els ordinadors han anat agafant major presència en el procés de producció i la possibilitat de disposar de programes capaços de presentar informació dels processos que executen els PLC's en temps real, va obrir un nou mercat al que les grans corporacions s'han llençat de ple amb programes potentíssims capaços de mostrar, governar processos com el PCS7 de Siemens o el Citec de Schneider.



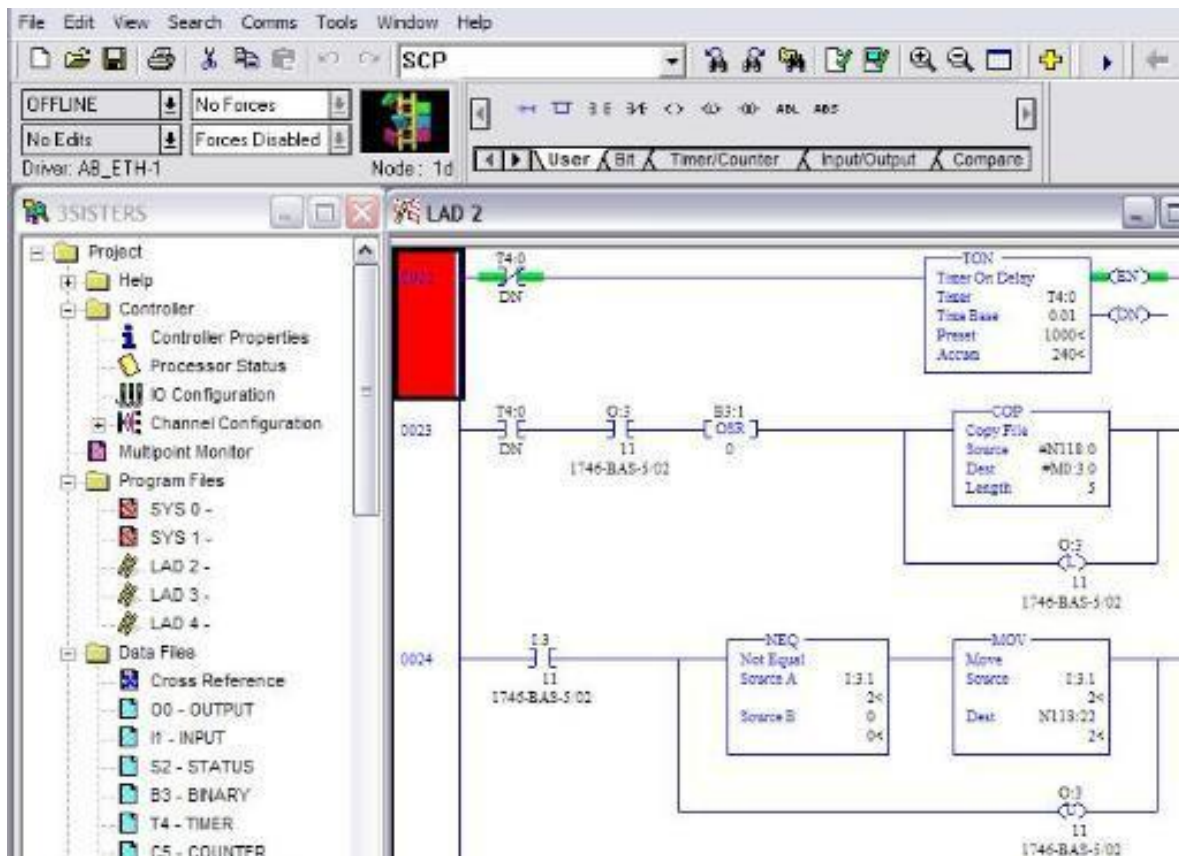
Pantalla d'un procés industrial amb PCS7



Pantalla d'un procés industrial amb Citect

Aquests canvis estan apropant als professionals de la informàtica al món de l'automatització industrial de manera cada vegada més evident. Els PLC's són servidors WEB, es programen en C i poden gestionar bases de dades.

El paradigma és que les empreses estan incorporant noves tecnologies als seus productes per a oferir millors prestacions, però, encara segueixen amb la política de tancament absolut dels seus protocols i codis.



Pantalla d'una programació Lader logic

A conseqüència d'aquesta aparent inconsistència, estan apareixent tímidament algunes plataformes basades en Linux per al control industrial que utilitzen les capacitats de gestió dels temps real que ofereix Linux per a crear Kernels emuladors de PLC.

El gran problema d'aquestes noves plataformes, és la fiabilitat del hardware i els desenvolupaments, motiu pel qual la seva implantació es molt petita i només s'utilitzen en situacions molt particulars.

Tot aquest escenari de fabricant de PLC's, SCADAS, gestors documentals, CAD's, etc. I la quantitat d'usuaris amb perfils diferents condueix a que la coordinació del treball i la capacitat de reutilitzar els desenvolupaments es considerin impossibles i quedin cenyits a gammes de productes, empreses instal·ladores o inclòs a grups de treball dins la mateixa empresa.

Actualment internet a dorçat a una ajor i millor comunicació entre els usuaris que reclament un esforç per estandaritzar maneres de programació, visualització, gestió documental i col·laboració.

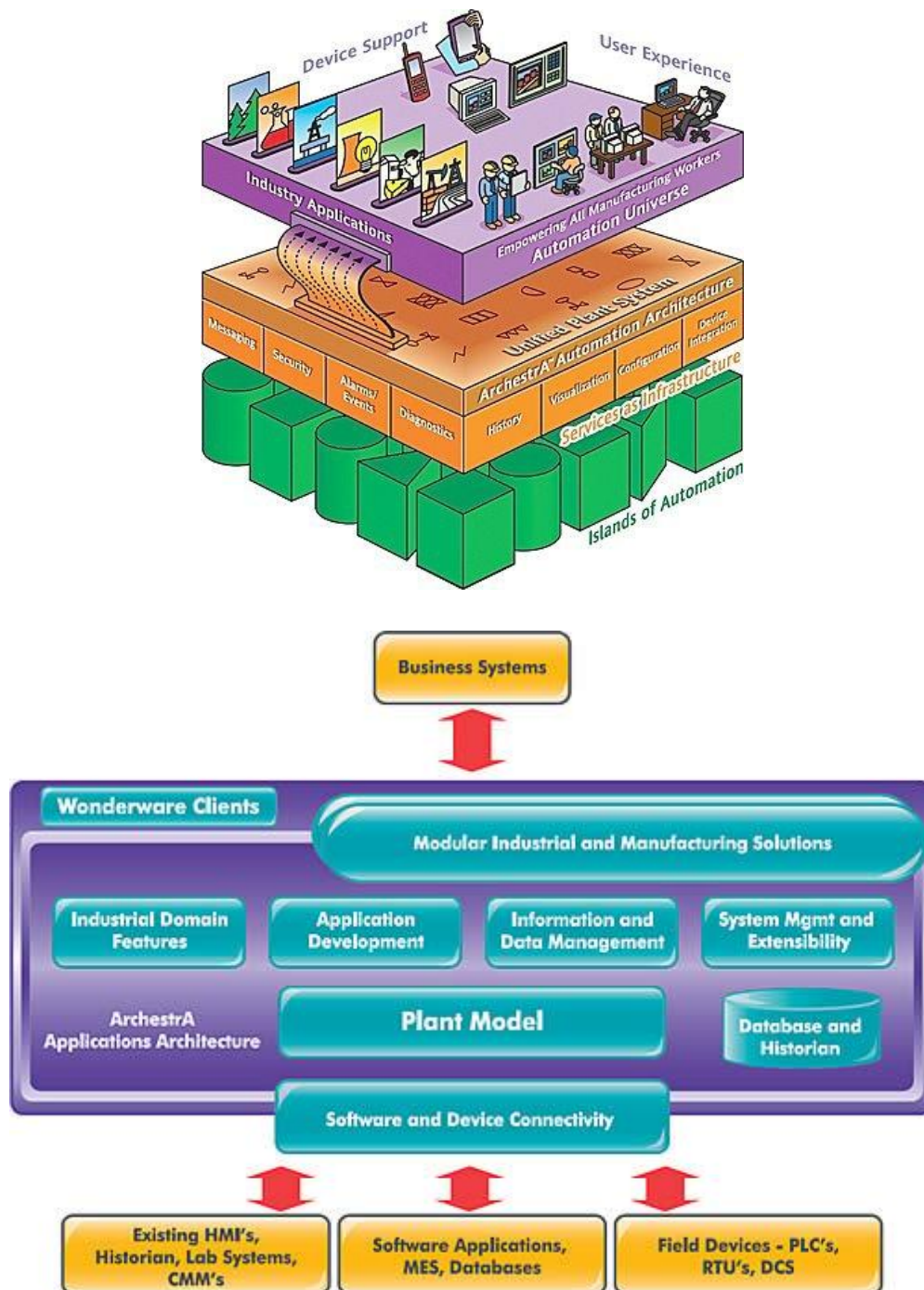
Empreses com SIEMENS han optat per crear gammes de productes verticals que poden cobrir totes les necessitats, desde la planificació, programació de PLC's/SCADA i reutilització i documentació. El problema d'aquests paquets es que son prohibitius i complexes, a part de propietaris.

Encara amb la presència en el mercat d'aquests paquets, no existeix una eina de gestió de projecte unificada per a entorns de control industrial. Aquestes eines comunament conegudes com a PLM(Project Live Manager) comporten la interrelació de les diferents parts d'un projecte com documentació, arquitectura de xarxa, models d'instal·lació, estandarització de programes, etc. Una de les millores en aquest moment es l'entorn EPlan-Platform, que pretén establir lligams tècnic-comercials amb les diferents plataformes de mercat i itnegrar capacitat de comunicació amb aquestes.



Un dels grans problemes d'EPlan-Platform és que va nèixer com un CAD-Elèctric i no té una filosofia de PLM de control de processos, sino, d'un “enginerring center” per als projectes de disseny de materials no de software.

Una de les millors aproximacions a una solució, és la proposada per la empresa americana de SCADAS Wonderware i el seu nou producte “Industrial Application Server”, qui ha efectuat alguns progressos en aquesta direcció, no obstant, molt lluny de la solució PLM completa.



En definitiva la perceptiva de disposar d'un PLM orientat a projectes de control industrial a la seva vertent de programació, amb mètodes de comparició de coneixement i capacitat de treballar amb diferents plataformes tant hardware (PLC), software (SCADA), com documental, es actualment un somni.

El nostre projecte es realitzarà tot en entorn web, totalment gratuït i podrà funcionar sobre sistemes operatius de lliure distribució amb software d'ús quotidià, fàcil d'utilitzar i sense tenir una instal·lació a màquina ja que es treballarà sota servidor, es podrà contemplar com un PLM (Project Live Manager)

per la creació i posterior manteniment de grans sistemes, des de la seva fase de disseny fins la seva implementació.

Avui en dia hi ha millors infraestructures de xarxa, a més altes velocitats, i nous dispositius mòbils, amb els que poder treballar en client. Això es un gran avantatge, del que podem treure un gran punt a favor respecte a qualsevol software anterior que es pugui trobar, tant de pagament com lliure.

Es vol aconseguir fer un projecte en mòduls, encarats a la utilització per part d'un usuari de nivell baix, utilitzant un net-view per qualsevol acció que es vulgui fer.

El mòdul d'objectes es converteix en un servidor virtualitzat, en el que no necessitem cap tipus de compra de llicències ja que treballa en un sistema operatiu lliure, en aquest cas em optat per CentOS. Aquest sistema ens donarà entrada a serveis de bases de dades i una interfície web per poder gestionar tots els serveis i la base de dades.

1.4 Aproximació històrica a la tecnologia

A llarg de la història, l'home ha desitjat que les màquines realitzin el treball més pesat y repetitiu de manera automàtica. Aquesta necessitat va sorgir de l'afany de dedicar més temps a tasques més complexes i per tant més gratificants.

Encara que en els principis de la història ja es van dissenyar sistemes automàtics per a tasques simples, com mantenir el nivell de líquid d'un dipòsit mitjançant una vàlvula regulada per una boia de flotació, no es considera que comença l'automatització industrial fins el 1750 amb l'inici de la revolució industrial.

1745: Màquines de teixit controlades per targetes perforades Jackard.

1817-1870: Màquines especials per al tall del metall.

1856-1890: Sir Joseph Whitworth emfatitza la necessitat de peces intercanviables.

1870: Primer torn automàtic, inventat per Christopher Spencer.

1940: Sorgeixen els controls hidràulics, neumàtics i electrònics per a màquines de tall automàtiques.

1945-1948: John Parsons comença investigació sobre control numèric.

1960-1972: Es desenvolupen tècniques de control numèric directe i manufactura computadoritzada.

Encara que inicialment la “intel·ligència” d'aquests dispositius es basava en combinacions de relés utilitzats com a portes lògiques, ràpidament es va observar que reprogramar els circuits de relés era una tasca àrdua i freqüentment es produïen errades.

A principis dels anys 70 van aparèixer els primers calculadors lògics, precursors del PLC's, en el sector de l'automoció en Estats Units. Aquests equips tenien l'avantatge que no feia falta tornar a cablejar relés per canviar el programa en execució ja que disposaven de consoles per entrar les condicions d'activació de les sortides en funció de les entrades.

En 1969, la Divisió Hydramatic de la General Motors va instal·lar el primer PLC per a reemplaçar els sistemes inflexibles alambrats fets servits llavors en les seves línies de producció.

Ja en 1971, els PLC's s'estenien a altres indústries i en els vuitanta ja els components electrònics van permetre un conjunt d'operacions en 16 bits, - comparats amb els 4 dels 70s -, en un petit volum, el que els va popularitzar en tot el món.

Encara que durant molts anys aquests PLC's (Programable Logic Controller) van ser patrimoni dels electricistes, amb el pas del temps aquests elements han evolucionat fins a convertir-se en petits ordinadors capaços d'efectuar un gran nombre de càlculs.

Un dels passos més importants en el control per PLC's va ser assumir les tasques dels controladors PID (Proporcional Integrador Derivador) per a regulació de processos continus. Això va ser possible gràcies a les grans capacitats de càlcul entregades pels microprocessadors moderns.

Actualment els PLC's s'estan llançant al món de la programació orientada a objectes, complexos bussos de comunicació, bases de dades, serveis WEB, etc. Es a dir, els PLC's ja no son patrimoni dels electricistes.

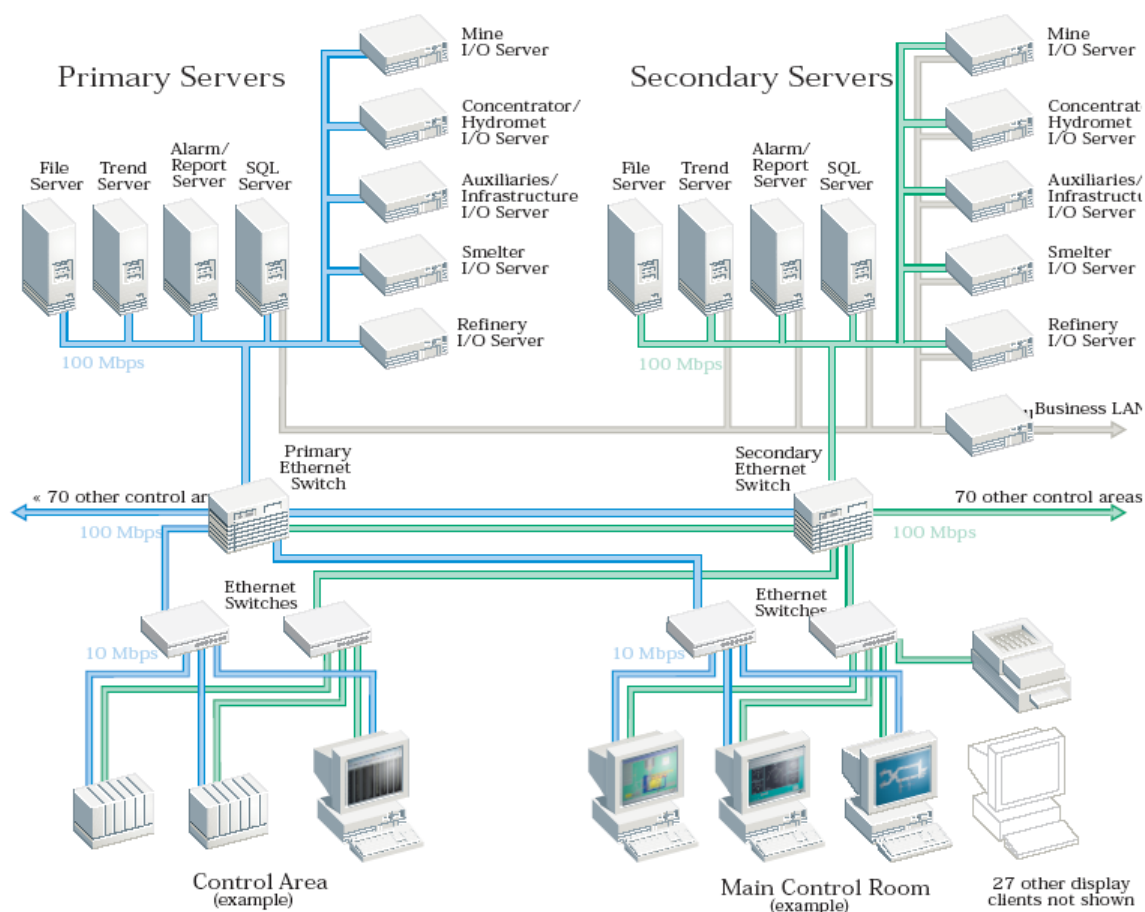
Amb l'aparició dels ordinadors a baix cost i les possibilitats gràfiques avançades van veure la llum els programes SCADA (Supervisory Control and Data Acquisition), aquests programes van permetre comunicar els PLC's i disposar de informació en temps real dels processos i estat de les variables internes de l'equip. Això va suposar una revolució en tots els entorns de processos industrials, ja que va permetre efectuar estudis molt precisos del comportament de les línies de producció mentre aquestes es produïen. Per primer cop la informació dels processos de producció es podia emmagatzemar, visualitzar i analitzar de manera precisa.

Actualment existeixen un gran nombre d'aplicacions d'aquestes característiques en el mercat, capaços de monitoritzar i governar instal·lacions de cents de milers de senyals en temps real.

L'aplicació més gran de SCADA en l'actualitat és el sistema de control de la mina Olympic Dam de Australia basat en el sistema Citec. Aquest sistema de controla més de 400.000 variables en temps real. L'equip que va desenvolupar el projecte comptava amb l'ajuda de és de 40 persones i treball durant 18 mesos.



Mina Olympic Dam, Australia



Arquitectura del projecto

Sense anar tan lluny, el sistema de control de la nova terminal T1 de l'aeroport de Barcelona controla 198.000 variables en temps real en una plataforma de més de 30 servidors y 1000 PLC's compartint informació.



Terminal T1 Aeropuerto de Barcelona

En projectes d'aquesta gran envergadura es quan la organització, coordinació i capacitat de gestió d'un entorn unificat per un PLM orientat a sistemes de control, es fa senzillament indispensable

1.5 Objectius previstos

El principal objectiu és poder realitzar un mòdul d'objectes amb totes les característiques necessàries per poder dissenyar un objecte al 100% per un futur tractament per un disseny de control SCADA, gratuït, evolutiu a la necessitat dels usuaris i modificable per part dels usuaris a les necessitats personals de cada un.

Ha de ser un mòdul molt senzill d'utilitzar i de gestionar, de tal manera que un usuari de nivell baix, pugui utilitzar totes les característiques del mòdul.

Integra-ho dins d'un llenguatge que ens permeti no tenir res instal·lat al client, així que em optat en un servei web amb una base de dades.

Fàcil de compartir dissenys entre els diferents usuaris, s'ha de poder veure elements publicats. Cada usuari tindrà una carpeta personal i una carpeta pública, on només es podran posar els elements finalitzats.

L'usuari ha de estar còmode treballant; l'usuari podrà organitzar els seus objectes dins de carpetes de la manera que vulgui per treballar a la seva manera i còmodament.

Per part dels administrador del servei informàtic, no han de tenir coneixements per arrancar el mòdul, tot estarà dins d'una màquina virtualitzada amb totes les característiques configurades, de tal manera que únicament s'haurà de arrancar el servidor preparat perquè els usuaris de la xarxa puguin començar a treballar.

Facilitat per migrar el servidor. Com el servidor estarà virtualitzat, per moure-ho de lloc, únicament s'haurà de moure la màquina virtual, de tal manera que és una tasca de 5 minuts.

A més ha de permetre poder compartir objectes de manera global, així que s'ha de crear un identificador únic universal, per això s'ha optat de crear una matrícula d'objecte la qual es forma a partir de la nostra IP versió 4, el nom de la nostra màquina i el temps en el que es crea l'objecte.

1.6 Fonaments teòrics

A continuació farem una explicació de cada una de les tecnologies utilitzades en el projecte, a més d'una explicació dels diferents acrònims i abreviacions:

1.6.1 Linux:

- És un sistema operatiu provinent de UNIX. Unix és un sistema operatiu robust, estable, multi usuari, multi tasca i multi plataforma amb una gran capacitat per a la gestió de la xarxa.
- Linux és un sistema operatiu compatible amb Unix. Té dos característiques molt peculiars que ho diferencien de la resta de sistemes que poden trobar en el mercat. La primera és que és lliure, això significa que no tenim que pagar cap tipus de llicència per el seu ús, la segona és que el sistema ve acompanyat del seu codi font. El sistema ho formen un nucli del sistema (kernel) més un gran número de programes/l·libreries que fan possible la seva utilització. Linux es distribueix sota “GNU Public License” amb que el codi font sempre és accessible.
- Característiques de LINUX:
 - Multi tasca: es poden executar diferents programes al mateix temps. Linux utilitza la multi tasca preventiva, que assegura que tots els programes que s'executen en un moment seran executats, sent el sistema operatiu l'encarregat de cedir el temps de microprocessador a cada un d'ells.
 - Multi usuari: poden estar treballant molts usuaris sota la mateixa màquina.
 - Multi plataforma: es pot utilitzar en moltes plataformes diferents, que caracteritzen a Linux sent molt compatible amb els sistemes, entre ells (386-, 486-. Pentium, Pentium Pro, Pentium II, Amiga y Atari, Alpha, ARM, MIPS, PowerPC y SPARC)
 - Multiprocessador: suporta processador de més d'un nucli
 - Protecció de la memòria entre processos.
 - Carga d'executables per demanda: Linux només llegeix del disc aquelles parts d'un programa que s'utilitzen en el moment.
 - Política de copia en escriptura per a poder compartir pàgines entre executables, això significa que diferents processos poden utilitzar la mateixa zona de la memòria per executar-se.

- Memòria virtual utilitzant paginació (sense intercanvi de processos complets a disc).
- La memòria es gestiona com un recurs unificat per als programes d'usuari y per al cachè de disc, de tal manera que tota la memòria lliure pot ser utilitzada per a cachè i reduïda quan s'executin programes grans.
- Llibreries compartides de càrrega dinàmica (DLL's) i llibreries estàtiques.
- Suport per molt teclats nacionals o adaptats.
- Servei de virtualització.
- Sistema d'arxius de fins a 4Tb i noms de 255 caràcters de longitud.
- Accés transparent a particions de Ms-DOS (particions OS/2 o FAT).
- Llegeix tots els format de CD-ROM.
- TCP/IP, incloent ftp, telnet, etc.
- LAN manager / Windows Native (SMB, SAMBA).
- Diversos protocols de xarxa inclosos al kernel: TCP, IPv4, IPv6, etc.

1.6.2 Apache:

- Acrònim de “patchy server”. És un servidor web de lliure distribució i codi obert. Actualment el més popular del mon, té els 50% dels servidors de tot el mon.
- S'executa sota múltiples sistemes operatius com Windows, Novell, NetWare, Mac Os X i els sistemes basats en UNIX.
- Característiques :
 - Suport per als llenguatges perl, python, tecl y PHP.
 - Mòduls d'autenticació.
 - Suport per SSL i TLS.
 - Permet la configuració de missatges d'errors personalitzats i negociació de contingut

- Ús d'Apache: s'utilitza principalment per servir pàgines web estàtiques i dinàmiques en la www. Apache és el servidor web més popular del sistema XAMP, junt a MySQL i els llenguatges PHP, Perl i Python.

1.6.3 MySql:

- És un sistema de gestió de base de dades (SGBD) multi usuari, multi plataforme i de codi obert (open source).
- MySQL és molt popular en aplicacions web, compatible en Windows, Mac i Linux, es combina habitualment amb PHP.
- Característiques de MySQL
 - Escrit en C i C++
 - Utilitza el llenguatge SQL per les consultes a les bases de dades.
 - Treballa sota una gran multitud de plataformes: [AIX](#), BSDi, [FreeBSD](#), HP-UX, [GNU/Linux](#), [Mac OS X](#), NetBSD, [Novell NetWare](#), OpenBSD, OS/2 Warp, QNX, SGI IRIX, [Solaris](#), SunOS, SCO OpenServer, SCO UnixWare, Tru64, [Microsoft Windows](#) (95, 98, ME, NT, 2000, XP , Vista i Windows7).

1.6.4 GISCMO:

Gestor Integrat de Sistemes de Control i Modelatge d'Objectes.

1.6.5 SCADA:

“Supervisory Control And Data Adquisition”, es tracta d'una aplicació software dissenyada per funcionar sobre ordinador de control de producció, donant comunicació amb dispositius de camp (controlador autòmats, autòmats programables, etc.), i controlant el procés de manera automàtica des de l'ordinador. També dona tota la informació que es genera en el procés productiu a diferents usuaris, tant al mateix nivell com d'uns altes nivells dins de l'empresa com control de qualitat, supervisió, manteniment, etc.

1.7 Estructura de la memòria

La nostra memòria s'organitzarà en set capítols, en els quals quedarà documentat el procés de desenvolupament del projecte, des de el seu estudi inicial, planificació, programació, creació d'una màquina virtual i entorn final de proves.

Rere aquest primer capítol introductori, es realitzarà l'estudi de viabilitat del projecte en el segon capítol, en el que es fa una breu descripció del projecte, estudi de la situació actual, lògica del sistema, recursos de hardware, software i humans, anàlisis de costos i es defineix com a viable o no.

2 Estudi de viabilitat

2.1 Tipologia i paraules clau.

- **Tipologia:** desenvolupament.
- **Paraules clau:** base de dades, modelatge i emmagatzematge d'objectes, “Model View”, fitxa de dades, control industrial.

2.2 Objecte

2.2.1 Situació actual

- Actualment no hi ha cap eina de software de lliure distribució i modificable que doni suport al disseny d'entorns de control industrial. No hi ha software capaç de fer el modelatge de sistemes de control, la creació de codi automatitzat per diferents sistemes, i que sigui intuïtiu i fàcil d'utilitzar.
- Els diferents programes estan dissenyats per treballar autònomament, és a dir, no estan preparats per tenir un enllaç entre ells, compartir recursos, configuracions, etc. Amb el GISCMO, tindrem tot en un mateix programa, que alhora aquest es podrà anar ampliant amb el temps, millorar característiques, ampliació de mòduls nous, etc.
- Fàcil gestió i manipulació.
- Fàcil manteniment, amplia informació a la xarxa sobre manteniment de bases de dades, i servidors Linux.
- Instal·lació i migració instantànies. Creat sobre un servidor Linux virtualitzat, fàcil de fer manteniment i snapshots, opció d'alta disponibilitat si es fa en clústers de servidors.
- Tot configurat des d'el principi, els administradors de la xarxa no han de configurar res del servidors des d'un punt de partida inicial.

2.2.2 Perfil d'usuari del projecte.

2.2.2.1 Perfils d'usuaris:

Nom	Descripció	Responsabilitat
Administrador	Administra els usuaris del mòdul	Creació d'usuaris del mòdul d'objectes.
Dissenyador	Dissenyador del disseny	Dissenyador d'objectes

El mòdul d'objectes tindrà 2 tipus d'usuaris: l'administrador i el dissenyador.

Al futur dissenyador no ha de tenir gran coneixements de la informàtica, el mòdul d'objectes GISCMO s'ha dissenyat de tal manera que sigui de fàcil aprenentatge i ús.

L'administrador tindrà un apartat per modelar objectes i a més tindrà una consola d'administració d'usuaris, a més

2.2.2.2 Project Team GISMO

- Modelatge de hardware
- Modelatge d'objectes
- Gestió i creació de diccionaris de senyals
- Gestió de projectes
- Mòdul de compilació per SCADA o PLC

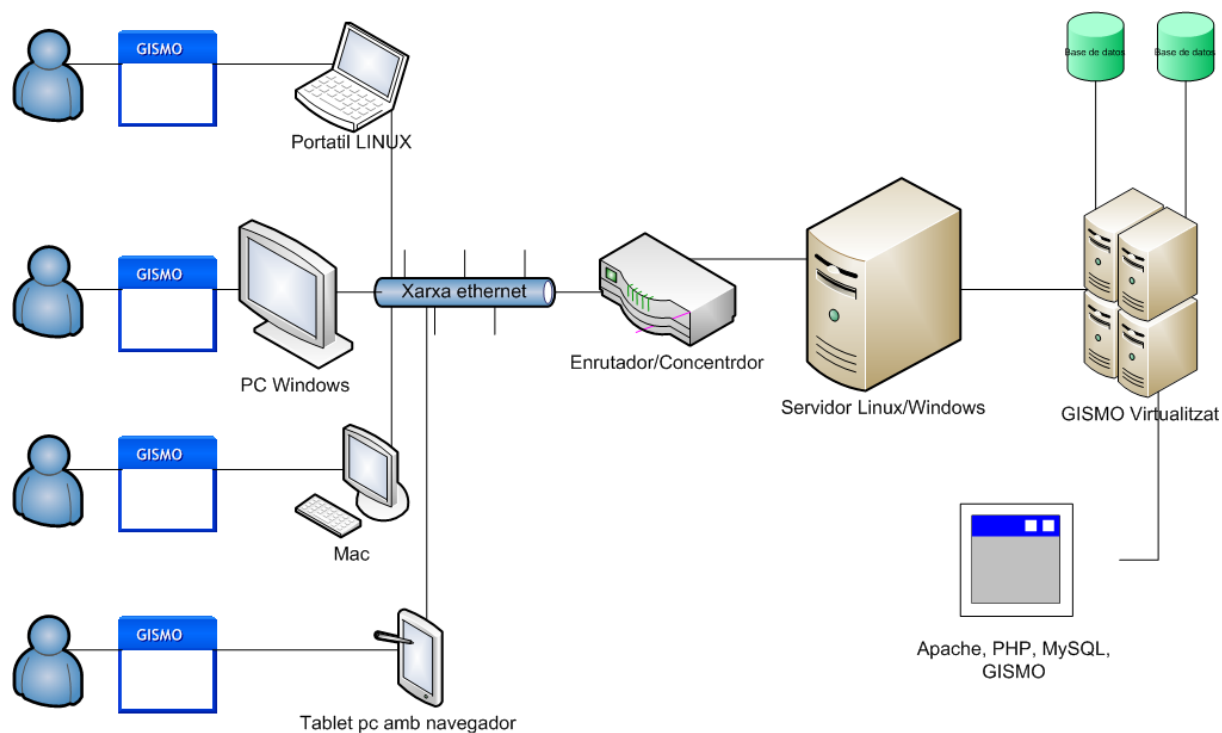
Nom	Descripció	Responsabilitat
A	Cap del projecte (CP)	Defineix, gestiona, planifica i controla el projecte.
B	Responsable mòdul de hardware	Anàlisi i disseny de la BBDD i model web, que compleixi els requeriments per modelar i emmagatzemar els components hardware d'una instal·lació d'un sistema de control.
C	Responsable mòdul d'objectes	Anàlisi i disseny de la BBDD i interfície web, tal que compleixi els requeriments per modelar i emmagatzemar el objectes propis d'una instal·lació d'un sistema de control
D	Responsable mòdul de diccionaris de senyals	Anàlisi i disseny de la BBDD i model web, que compleixi els requeriments per poder gestionar, crear i modificar els senyals que puguin tenir el hardware i els objectes dins la instal·lació.
E	Gestió de projectes	Mòdul que permet gestionar el projectes que facin viable la instal·lació d'un sistema de control.
F	Responsable del mòdul de compilació	Serà capaç de compilar el sistema de control amb els objectes, hardware i senyals digitals d'un sistema de control.

2.3 Objectius

Nom	Descripció
I	Creació d'un entorn operatiu web senzill
II	Seguretat a les comptes d'usuaris amb codificació de la paraula de pas amb M5
III	Creació d'una estructura jeràrquica en arbre dels objectes
IV	Creació, modificació, i eliminació dels elements de cada usuari
V	Inserció, eliminació i modificació de senyals provinents de diccionaris del mòdul de diccionaris
VI	Inserció, modificació i eliminació de característiques amb possibilitat d'obligació de característica
VII	Llista d'objectes públics
VIII	Llista d'objectes propis per a cada usuari
IX	Creació d'una matrícula única universal per a cada objecte
X	Còpia d'objectes públics per ser modificats per el dissenyador
XI	Gestió d'usuaris únicament per administradors
XII	Creació d'un sistema virtualitzat amb un sistema operatiu lliure i el mòdul d'objectes integrat amb tots els serveis activats (PHP, APACHE, MYSQL)

2.4 Descripció del sistema

2.4.1 Lògica del sistema



2.5 Recursos

2.5.1 Recursos humans:

Nom	Descripció	Responsabilitat	Hores
H1	Cap de projecte	Defineix, gestiona, planifica i controla el projecte, supervisa la feina de l'alumne, avalua el projecte...	17
H2	Alumne programador/dissenyador	Documentació, gestió, creació i definició del projecte.	425

2.5.2 Recursos hardware

Recurs	Nom	Descripció
HW1	Portàtil HP Pavilion	Portàtil HP, amb processador AMD Athlon 64 doble nucli, 4Gb de RAM
HW2	Swich	Habilita la connexió amb la resta de la xarxa.
HW3	Disc dur extern	Disc dur extern de 150Gb per emmagatzemar el treball 150Gb
HW4	Cable de xarxa RJ45	Cable per poder establir la connexió amb el router
HW5	Llibreta	Llibreta per dur a terme el treball a mà(pseudocodi, diagrames, etc.)

2.5.3 Recursos software

Recurs	Nom	Descripció
SW1	Windows 7 versió universitària	Sistema operatiu Windows 7
SW2	CentOS 5.3	Sistema operatiu CentOS amb el que es crea la màquina virtual
SW3	Microsoft office 2007	Software per crear la documentació del projecte
SW4	Microsoft Visio 2007	Software per crear els organigrames de la memòria
SW5	MySQL	Servei MySQL de lliure distribució
SW6	Mozilla Firefox 3.5.9	Navegador web gratuït
SW7	PHP	Servei PHP per el servidor Linux
SW8	Apache	Servei Apache per crear el webserver
SW9	Putty	Programa per establir una connexió SSH amb consola de comandes de linux, per treballar remotament amb el servidor linux
SW10	Webmin	Servei Web per l'administració i gestió Linux i molts dels mòduls instal·lats
SW11	VmWare	Software de virtualització amb el que es crea el servidor Linux
SW12	WinSCP	Client SSH per windows que permet connectar a Linux activant SSH per transferir arxius
SW13	Microsoft Project	Aplicació per crear el diagrama de Gantt amb les tasques i els recursos

2.5.4 Recursos amb costs

Recurs	Cost
Connexió a internet	0 (proporcionat per la universitat)
Electricitat	0 (proporcionat per la universitat)

2.6 Planificación del proyecto











2.6.1 Planificación inicial

Tal i com em pogut fer la planificació del projecte, el desenvolupament es realitzarà per fases de gran durada per la seva complexitat, així que s'ha decidit fer la planificació seguint un mòdel seqüencial de les diferents tasques, ja que necessitem planificar bé el projecte per que fent una tasca tinguem tot el necessari per crear la següent tasca.

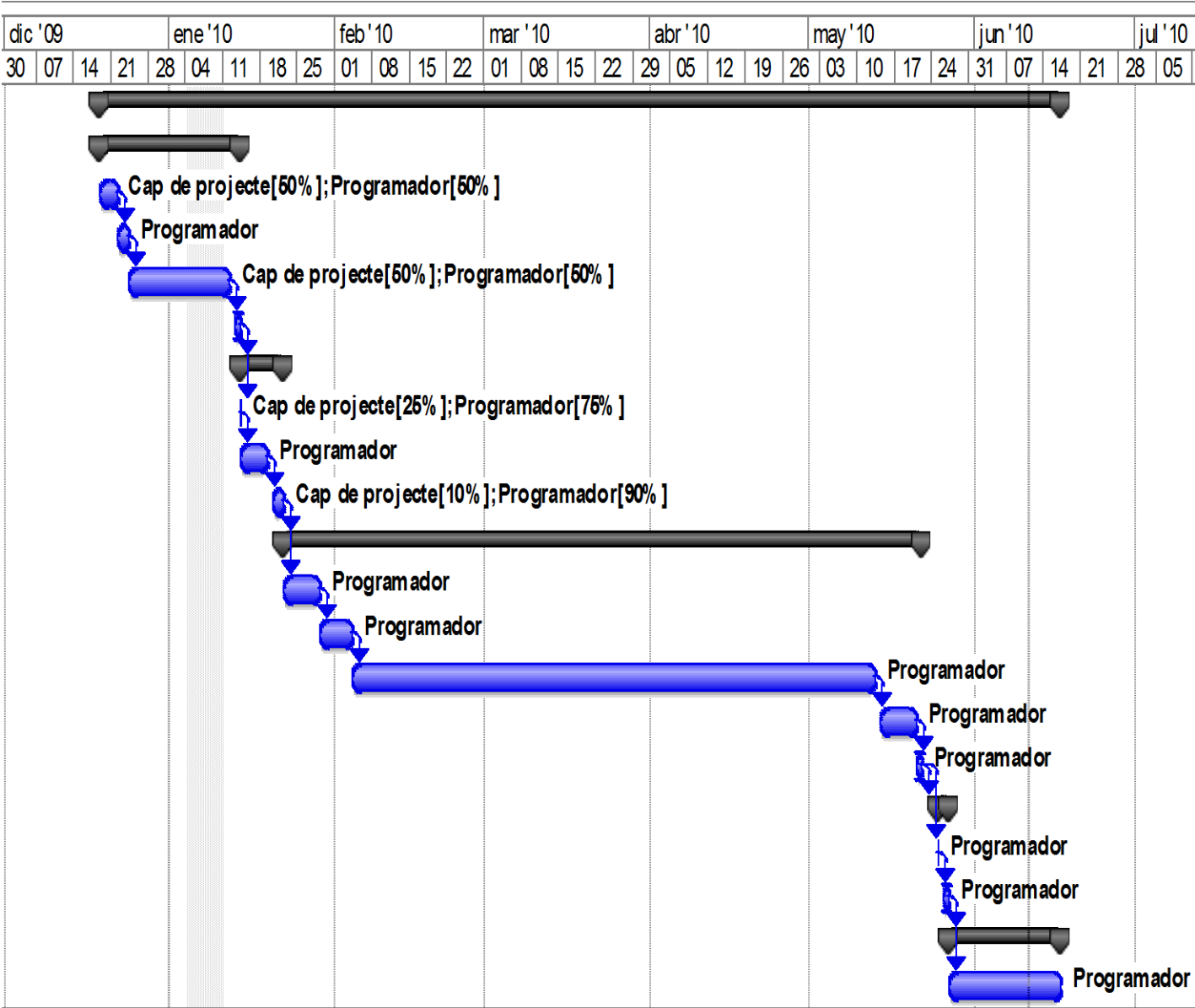
Això ens permetrà desenvolupar el projecte de manera lineal, mantenint aquest ordre planificat amb el cap de projecte, partint des d'una base de creació de les bases de dades, creació del codi PHP amb la interfície web, creació d'un sistema virtualitzat en entorn Linux amb VmWare i el testeig de cada una de les parts.

El fet de conèixer els requeriments des d'un inici i no tenir riscos significatius ja que utilitzem eines prou conegudes de les que podem tenir molta informació, ens facilita la tasca de seguir una bona planificació.

2.6.2 Descripció de les tasques

	Nombre de tarea	Duración	Comienzo	Fin	Predecesoras	Nombres de los recursos
	 GISCMO Mòdul d'objectes	97,5 días	vie 18/12/09	mié 16/06/10		
	 Recopilació d'informació	7,75 días	vie 18/12/09	mié 13/01/10		Cap de projecte[50%];Pi
	Exemples	2 horas	vie 18/12/09	lun 21/12/09		Cap de projecte[50%];Prog
	Documentació prèvia	8 horas	lun 21/12/09	mié 23/12/09	3	Programador
	Funcionalitats GISCMO	16 horas	mié 23/12/09	lun 11/01/10	4	Cap de projecte[50%];Prog
	Informació sobre la tecnologia a utilitzar	5 horas	mar 12/01/10	mié 13/01/10	5	
	 Diseny de l'aplicació	5,5 días	mié 13/01/10	jue 21/01/10	6	
	Anàlisi funcionalitats	2 horas	mié 13/01/10	mié 13/01/10	6	Cap de projecte[25%];Prog
	Disseny de la base de dades	10 horas	mié 13/01/10	lun 18/01/10	8	Programador
	Disseny estructura aplicació	10 horas	mar 19/01/10	jue 21/01/10	9	Cap de projecte[10%];Prog
	 Implementació programació	73 días	jue 21/01/10	vie 21/05/10	10	
	Creació del sistema virtualitzat	18 horas	jue 21/01/10	jue 28/01/10	10	Programador
	Creació i configuració de la BBDD	12 horas	jue 28/01/10	mié 03/02/10	12	Programador
	Programació de l'aplicació	240 horas	mié 03/02/10	mié 12/05/10	13	Programador
	Programació de la interfície dels usuaris	20 horas	jue 13/05/10	jue 20/05/10	14	Programador
	Introducció de dades a la BBDD	2 horas	jue 20/05/10	vie 21/05/10	15	Programador
	 Proves	2,5 días	lun 24/05/10	mié 26/05/10	16	
	Realització proves	4 horas	lun 24/05/10	lun 24/05/10	16	Programador
	Correcció d'errors	6 horas	mar 25/05/10	mié 26/05/10	18	Programador
	 Memòria	8,75 días	mié 26/05/10	mié 16/06/10	19	
	Redacció memòria projecte	35 horas	mié 26/05/10	mié 16/06/10	19	Programador

2.6.3 Diagrama de Gantt



2.7 Anàlisi de costos

2.7.1 Cost de la càrrega de treball o recursos humans

Ja que estem en el cas del projecte GISCMO, els recursos cap de projecte i el programador/dissenyador tenen cost 0€. Aleshores encara que la durada del projecte és de 97,5 dies, el cost dels nostres recursos es de 0€ en total.

2.7.2 Cost dels recursos de hardware

Recurs	Nom	Cost
HW1	Portàtil HP Pavilion	660€
HW2	Switch	Inclòs amb la línia d'ADSL
HW3	Disc dur extern 150Gb	79€
HW4	Cable de xarxa RJ45	0€
HW5	Llibreta	1,50€

2.7.3 Cost dels recursos de software

Recurs	Nom	Cost
SW1	Windows 7 versió universitària	Proporcionat per la universitat
SW2	CentOS 5.3	Gratuït
SW3	Microsoft office 2007	Proporcionat per la universitat
SW4	Microsoft Visio 2007	Proporcionat per la universitat
SW5	MySQL	Gratuït
SW6	Mozilla Firefox 3.5.9	Gratuït
SW7	PHP	Gratuït
SW8	Apache	Gratuït
SW9	Putty	Gratuït
SW10	Webmin	Gratuït
SW11	VmWare	Gratuït
SW12	WinSCP	Gratuït
SW13	Microsoft Project	Proporcionat per la universitat

2.7.4 Avaluació de riscos

Les característiques del nostre projecte impliquen uns riscos necessàriament a tenir en compte, de tal forma que puguem preveure una situació de perill, en les que l'aplicació no funcioni adequadament o que tingui pèrdues de funcionalitats.

A continuació tenim una llista amb els factors que ens podria donar problemes amb el correcte funcionament de l'aplicació.

- Incompatibilitat amb Internet Explorer, el projecte està encarat a utilitzar el navegador gratuït Mozilla Firefox

- Problemes en els serveis de xarxa

- Insuficient espai en disc

- Manteniment davant caigudes del servidor i pèrdua de dades

Per poder disminuir el mínim els riscos de manteniment i la pèrdua de dades, em optat per fer un sistema virtualitzat amb VMWare, de tal manera que tenim a la nostra disposició tot el conjunt d'avantatges que tenim gràcies a la virtualització(per exemple, capacitat d'snapshoots, canvis de software, migració entre sistemes, fàcil de fer còpies de seguretat, portabilitat, etc.)

No obstant, els administradors seran els únics amb privilegis per poder crear o esborrar comptes d'usuari i del manteniment del sistema virtualitzat.

2.8 Conclusions

La primera conclusió es la viabilitat del projecte, segons el nostre estudi, podem concloure que “Es viable” sempre tenint en compte els riscos.

3 Anàlisi del projecte

3.1 Introducció

En aquest capítol es farà una descripció detallada del comportament de l'aplicació, es realitzarà un anàlisi en profunditat de les funcionalitats del portal.

L'anàlisi inclou un apartat dels requeriments funcionals els quals, mitjançant casos d'ús i diagrames de seqüència, es descriurà el funcionament de l'aplicació.

A l'estudi previ es va indicar que l'aplicació tindria usuaris i administradors, als diferents diagrames observarem les diferències entre ells.

Els usuaris només tindran una pantalla principal amb una breu introducció del mòdul de modelatge d'objectes, una pantalla de benvinguda i la opció d'anar a la interfície del mòdul de modelatge d'objectes. Els administradors, a més, tindran un menú per tractar als usuaris, i fer la gestió d'aquests.

En quant a la interfície del modelatge d'objectes, l'administrador es exactament igual que un usuari normal i tant un com l'altre tenen la opció de creació, modificació, eliminació i publicació d'objectes amb el conjunt de característiques i senyals que ha de tenir els objectes.

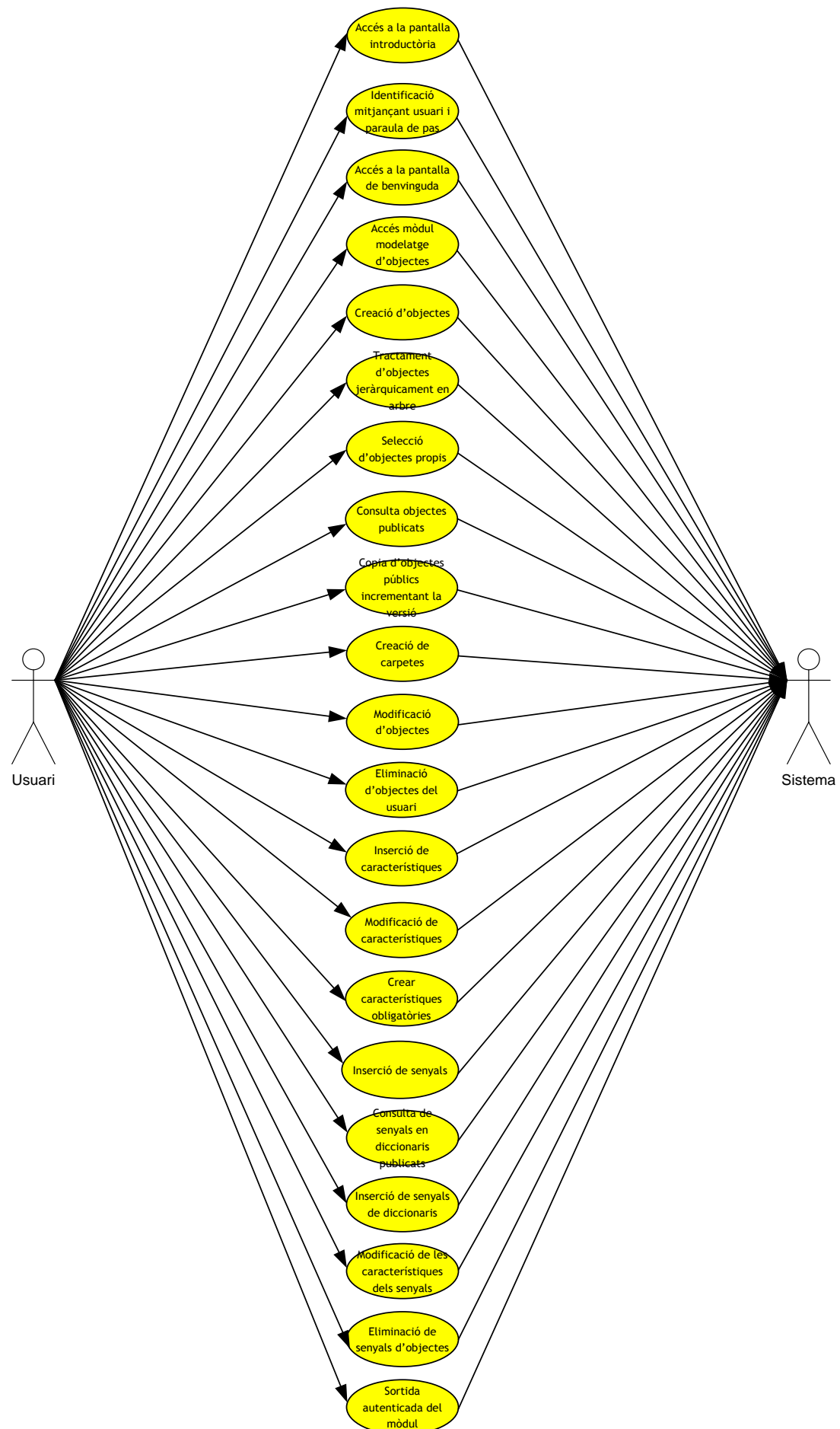
3.2 Requeriments funcionals

L'estudi es realitzarà mostrant els requeriments funcionals de cada tipus d'usuari, mitjançant diagrames de casos d'ús. Posteriorment es descriuran un conjunt de casos rellevants utilitzant diagrames de seqüència. Ja que l'administrador pot fer totes les mateixes funcions que un usuari, més el tractament dels usuaris, no s'indicaran aquells casos d'ús que faci l'usuari de tal manera que el gràfic no sigui massa extens i sigui més clar.

3.2.1 Diagrames de casos d'ús

3.2.1.1 *Requeriments funcionals d'un usuari*

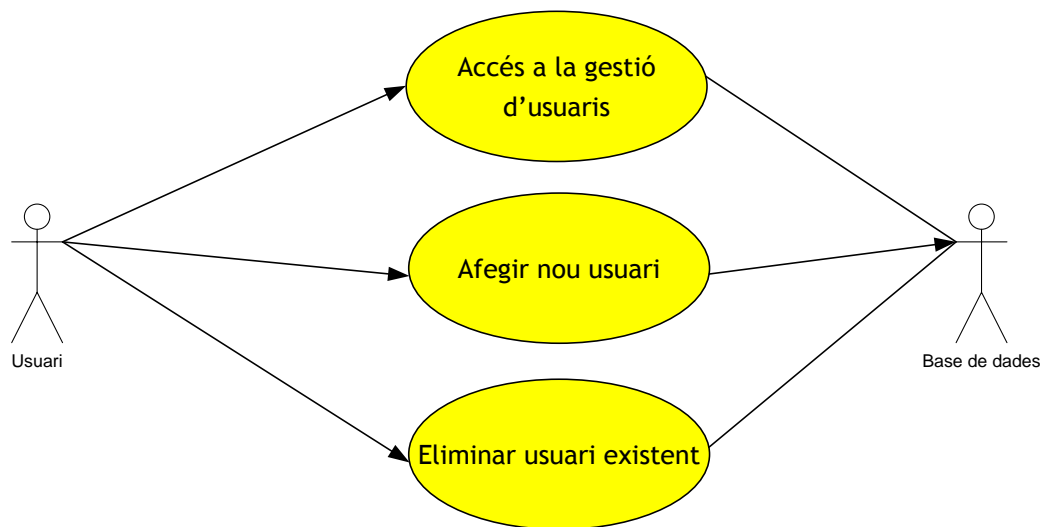
- Accés a la pantalla introductòria
- Identificació mitjançant usuari i paraula de pas
- Accés a la pantalla de benvinguda
- Accés mòdul modelatge d'objectes
- Creació d'objectes
- Tractament d'objectes jeràrquicament en arbre
- Selecció d'objectes propis
- Consulta objectes publicats
- Còpia d'objectes públics incrementant la versió
- Creació de carpetes
- Modificació d'objectes
- Eliminació d'objectes del usuari
- Inserció de característiques
- Modificació de característiques
- Crear característiques obligatòries
- Inserció de senyals
- Consulta de senyals en diccionaris publicats
- Inserció de senyals de diccionaris
- Modificació de les característiques dels senyals
- Eliminació de senyals dels objectes
- Sortida autenticada del mòdul



3.2.1.2 Requeriments funcionals d'un administrador

Les mateixes que un usuari més:

- Accés a la gestió d'usuaris
- Afegir nou usuari
- Eliminar usuari existent



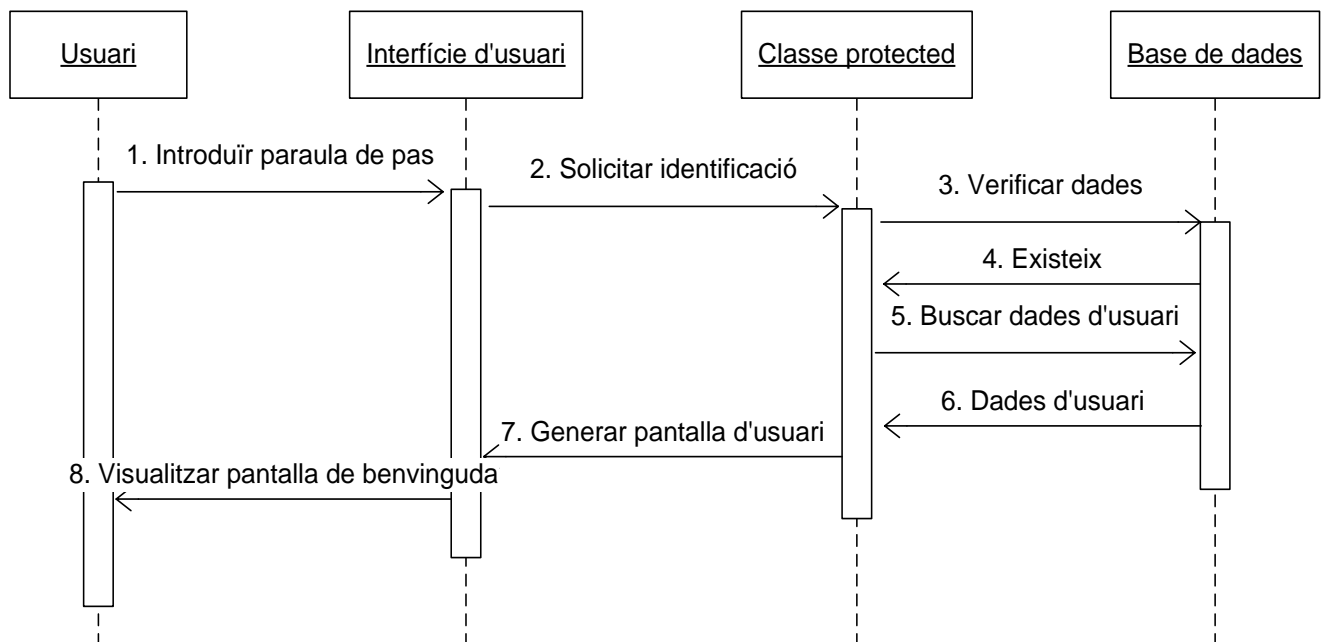
3.2.2 Diagrames de seqüència

A continuació, mitjançant diagrames de seqüència, es descriuen una selecció de casos d'ús que per el seu funcionament i característiques es consideren més rellevants.

3.2.2.1 Cas d'ús: Identificar-se

- Actors: usuari administrador, usuari
- Descripció: un usuari s'identifica en el sistema.

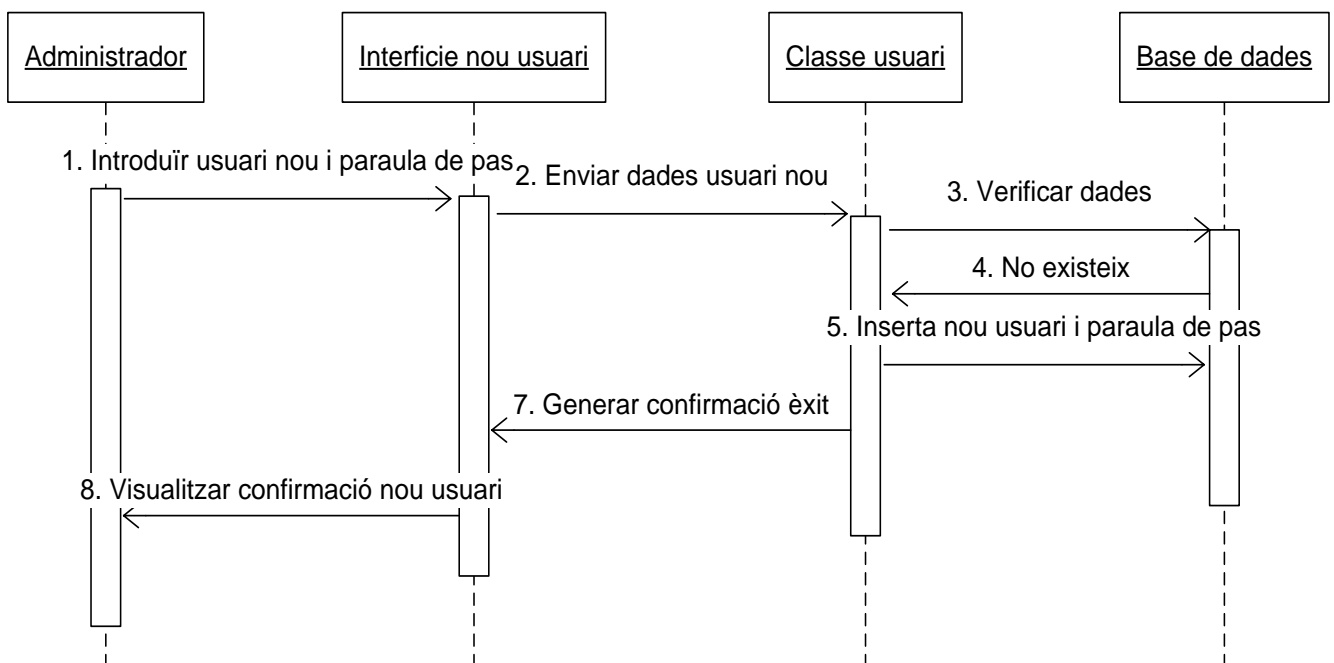
ACTOR	Sistema
1. L'usuari introdueix el seu nom d'usuari i la paraula de pas.	2. La classe registro comprova si l'usuari existeix a la base de dades, comprova si es administrador o usuari, i torna el resultat a l'usuari.
3. L'usuari/administrador es redirigit a la seva pàgina de benvinguda.	



3.2.2.2 Cas d'ús: crear usuari

- Actors: administrador
- Descripció: l'administrador, desde el menú de gestió d'usuaris, pot crear usuaris.

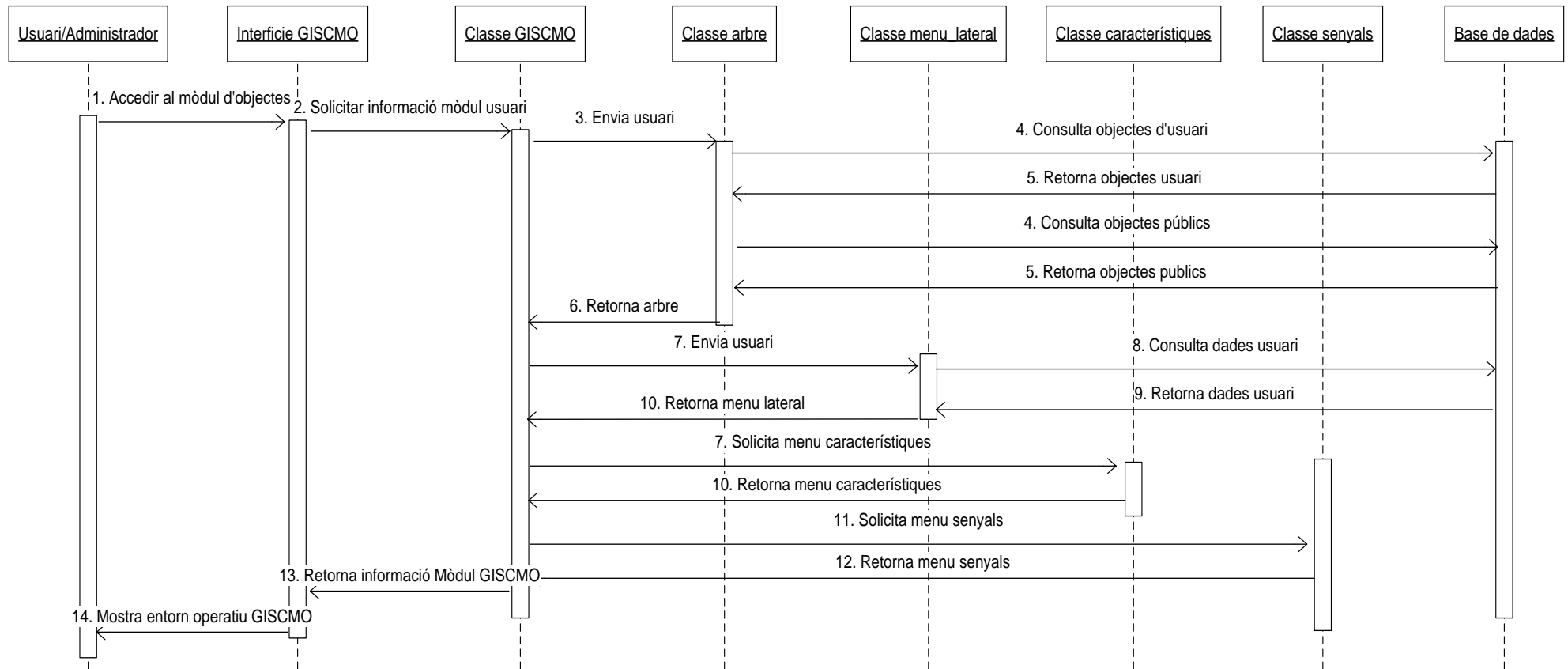
ACTOR	Sistema
1. L'administrador introdueix un nou nom d'usuari i la seva paraula de pas.	2. La classe usuari comprova si l'usuari ja existeix, si no existeix, crea un nou usuari.
3. L'administrador veu notificada que l'usuari s'ha creat.	



3.2.2.3 Cas d'ús: accedir a l'entorn operatiu Mòdul d'objectes

- Actors: administrador/usuari
- Descripció: l'administrador, des del menú de gestió d'usuaris, pot crear usuaris.

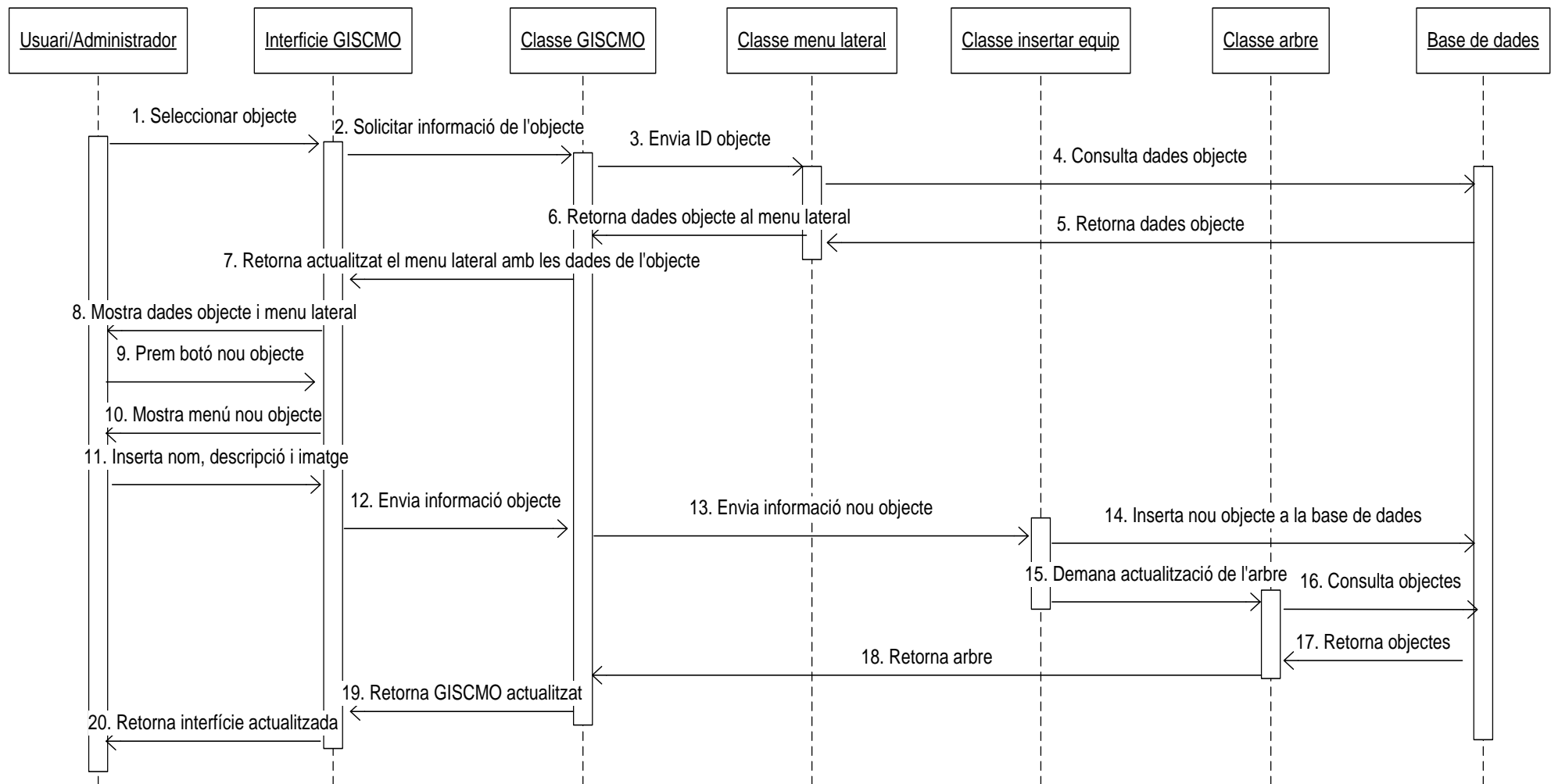
ACTOR	Sistema
<p>1. L'usuari accedeix a l'entorn operatiu del modelatge d'objectes</p> <p>6. L'usuari veu l'entorn operatiu del modelatge d'objectes en el seu estat inicial.</p>	<p>2. La interfície GISCMO crida a la classe GISCMO.</p> <p>3. La classe GISCMO li envia l'usuari a la classe arbre, la classe arbre consulta la informació de l'usuari amb la base de dades i tots els seus objectes. També consulta tots els objectes publicats.</p> <p>4. La classe GISCMO envia l'usuari a la classe menu_lateral, el menú lateral consultarà amb la base de dades la informació de l'usuari.</p> <p>5. La classe GISCMO carrega el menú característiques i el menú senyals cridant a característiques i senyals.</p>



3.2.2.4 Cas d'ús: nou equip

- Actors: administrador/usuari
- Descripció: l'usuari escull un objecte de l'arbre i crea un nou objecte amb un nom, una descripció i una imatge, fill del objecte seleccionat.

ACTOR	Sistema
1. L'usuari selecciona un objecte de l'arbre d'objectes	2. La interfície GISCMO crida a la classe GISCMO. 3. La classe GISCMO carrega el panell lateral amb les dades de l'objecte fent una consulta a la base de dades. 4. Mostra a l'usuari les dades de l'objecte.
5. L'usuari prem "nou equip"	6. La interfície GISCMO mostra la finestra nou equip
7. L'usuari introdueix un nom per al nou equip, una descripció i una imatge	8. La interfície GISCMO envia les dades a la classe insertar_nou_equip que les inserta a la base de dades 9. Es recarrega l'arbre amb el nou objecte insertat
10. L'usuari veu una confirmació que s'ha insertat correctament i l'arbre recarregat amb el nou objecte	



3.3 Requeriments no funcionals

Els requeriments no funcionals fan referència a restriccions que necessitem o que han sorgit per successos previstos o imprevistos que afecten al disseny final.

Aquí s'estudiarà, en primer lloc les restriccions de disseny i en segon lloc s'enumeraran els objectius del disseny.

3.3.1 Restriccions de disseny

3.3.1.1 *ID única universal*

Ja que volem que els objectes creats a partir del modelatge d'objectes es puguin compartir globalment, tenim que poder crear una matrícula d'objecte universal, garantint que no es podrà repetir, o que serà improbable, de tal manera que els objectes publicats no puguin crear errors dins la base de dades.

Així que a l'hora de crear un objecte tenim que fer aquesta matrícula per assignar-la com a identificador únic del objecte.

3.3.1.2 *Recuperació i fiabilitat davant errors*

Per poder assegurar que no es produiran errors, el sistema haurà de tenir un mecanisme de confirmació davant a diverses accions de l'usuari, com modificar objectes, eliminar objectes, eliminar usuaris, crear usuaris nous, etc. de tal manera, que l'aplicació no modifiqui o elimini cap dada de manera equivocada, per culpa d'un error de l'usuari a l'interactuar amb el mòdul de modelatge d'objectes i la interfície de creació d'usuaris.

L'entorn operatiu a més, treballarà amb la base de dades i amb arxius físics, que seran les imatges dels objectes, que estaran ubicats a les carpetes especificades, de tal manera que també s'haurà de comprovar que no hi hagi imatges amb noms repetits que puguin ocasionar incoherències.

Així que serà necessari que a la programació es tingui en compte que una mala inserció, modificació o eliminació a la base de dades, podria crear

incoherències, podríem perdre objectes realment existents a la base de dades, i podríem ocupar espai innecessari per els residus que podem generar si no fem una bona eliminació dels objectes.

3.3.1.3 Seguretat

Degut a les característiques del mòdul de modelatge d'objectes, i la opció de poder tenir múltiples usuaris, ha estat necessari implementar un tipus usuari i un tipus administrador.

Serà necessari assegurar que els usuaris no poden accedir a la gestió d'usuaris, i que tampoc no poden saber les contrasenyes dels administradors, per això, les contrasenyes han estat codificades amb M5.

Si un usuari intenta accedir a un apartat a on no pot tenir accés, o una persona que no és ni usuari vol accedir al mòdul, serà redirigit directament a la pantalla inici, sent aquesta la única que pot visualitzar.

3.3.2 Objectius de disseny

3.3.2.1 Facilitat de manteniment i ús

Donat que l'aplicació tindrà usuaris i administradors, que no han de tenir coneixements informàtics previs, es fa necessari realitzar un sistema que ens permeti una interacció molt senzilla per qualsevol usuari i intuïtiva.

3.3.2.2 Programació estructurada

S'ha de programar una aplicació que ens permeti fer actualitzacions o canvis amb el pas del temps, ja sigui per noves necessitats o per carències en el codi, per això el codi esta fet de manera estructurada amb diferents funcions independents fàcils d'assimilar per els programadors.

3.3.2.3 Accessibilitat

S'ha de dissenyar un sistema de menús de manera que l'usuari pugui accedir de manera ràpida i senzilla, i poder fer les funcions de l'entorn operatiu de manera intuïtiva. Per això el mòdul consta d'un menú amb botons fàcils d'entendre, i amb un sistema de finestres tipus sistema operatiu, que no serà nou per a l'usuari. També es tindrà un apartat de característiques i un de senyals utilitzant imatges fàcils d'entendre que indiquin prou clarament que fan els diferents botons.

3.3.2.4 Creació de finestres dinàmiques

S'ha volgut fer la programació de manera dinàmica, de tal manera que la informació s'actualitzi constantment en el moment de seleccionar un objecte. Disposarem de l'esmentat sistema de finestres que en tot moment tindrà un botó superior esquerre per tancar la finestra oberta, i també un botó cancel·lar fent la mateixa funció, també ens podem trobar un botó acceptar, per les finestres que puguin completar una acció. A més, aquestes finestres tindran la capacitat de poder moure's dins l'entorn del modelatge d'objectes, així deixant interactuar a través de l'arbre i poder potser copiar informació d'un altre objecte. És un sistema semblant a un sistema operatiu convencional al que els usuaris d'avui en dia estan acostumats a treballar.

3.3.2.5 Manteniment, seguretat del sistema i instal·lació

Per tal de garantir major seguretat de tot el sistema i el conjunt de totes les dades, s'ha creat dins un entorn virtualitzat, el que ens permetrà una fàcil migració, una fàcil còpia sencera del sistema i un fàcil manteniment, a més de la possibilitat de fer un sistema d'alta disponibilitat assegurant que el sistema sempre estarà actiu i serà molt difícil tenir una pèrdua de dades per culpa del hardware, així, que serà molt fiable davant de desastres.

4 Disseny

4.1 Introducció

En aquest capítol s'analitzarà com donar resposta a tots els requeriments que s'han tret del capítol anterior. Així doncs, d'ell dependrà en gran part l'èxit final del projecte, així que es definiran totes les parts que esmentat al capítol anterior i que defineixen el portal.

En aquests capítol, estudiarem les dos parts més importants del projecte, que és el mòdul de modelatge d'objectes i la seva base de dades, a on es guardarà tota la informació per al correcte funcionament del projecte i les aplicacions. Com es detallat anteriorment, l'entorn operatiu es farà de manera que un usuari sense coneixements informàtics pugui utilitzar el mòdul de modelatge d'objectes de manera fàcil i intuïtiva.

De la programació de l'entorn operatiu podem destacar el llenguatge utilitzat, en aquest cas es utilitza el llenguatge PHP per a la codificació i implementació de tot l'entorn operatiu. També s'ha utilitzat però en menor quantitat el Javascript, que ens permeten modificar propietats dels objectes del codi html, també es utilitza una funció AJAX que ens permet la càrrega dinàmica dels diferents mòduls de l'entorn operatiu.

Després de definir els dos llenguatges més utilitzats en el mòdul de modelatge d'objectes, tenim que dir, que el projecte i les diferents funcions són dependents de la base de dades, no podria fer les diferents funcions sense un lloc on emmagatzemar les dades dels diferents objectes, com el nom, descripció, característiques, etc. A més, el mòdul de modelatge d'objectes dins el projecte GISCMO, es recolza en el mòdul de creació i gestió de diccionaris de senyals dels objectes, el qual ens dona encara més característiques dels objectes.

Sempre tenim que tenir en compte que el mòdul de modelatge d'objectes es pot dur a terme de manera independent. No obstant, s'ha de tenir en compte que la seva funció verdadera és ser una part del projecte GISCMO. Així que hi ha

d'haver una coherència entre els mòduls, i s'han de conèixer també les característiques dels altres mòduls de tal manera que es pugui garantir que es podran comunicar sense problemes. Per la part que ens toca del mòdul de gestió de diccionaris per les senyals, em implementat diferents diccionaris que tenen la mateixa estructura que els reals del mòdul de gestió de diccionaris de senyals, garantint un bon funcionament amb el mòdul.

4.2 El model relacional

4.2.1 Introducció

El model relacional per a la gestió d'una base de dades és un model de dades basat en la lògica de predicat i en la teoria de conjunts. És el model més utilitzat en l'actualitat per modelar problemes reals i administrar dades dinàmicament. Després de ser postulades les seves bases en 1970 per Edgar Frank Codd, dels laboratoris IBM a San José (Califòrnia), no va trigar a consolidar-se com un nou paradigma en els models de base de dades.

La seva idea fonamental és l'ús de «relacions». Aquestes relacions podrien considerar-se en forma lògica com a conjunts de dades anomenades «tuples». Malgrat que aquesta és la teoria de les bases de dades relacionals creades per Edgar Frank Codd, la majoria de les vegades es conceptualitza d'una manera més fàcil d'imaginar, això és, pensant en cada relació com si anés una taula que està compostes per *registres* (cada fila de la taula seria un registre o *tupla*), i *columnes* (també cridades *camps*).

4.2.2 Descripció

En aquest model totes les dades són emmagatzemats en relacions, i com cada relació és un conjunt de dades, l'ordre en el qual aquests s'emmagatzemin no té rellevància (a diferència d'altres models com el jeràrquic i el de xarxa). Això té el considerable avantatge que és més fàcil d'entendre i d'utilitzar per un usuari no expert. La informació pot ser recuperada o emmagatzemada per mitjà de consultes que ofereixen una àmplia flexibilitat i poder per administrar la informació.

Aquest model considera la base de dades com una col·lecció de relacions. De manera simple, una relació representa una taula que no és més que un conjunt de files, cada fila és un conjunt de camps i cada camp representa un valor que interpretat descriu el món real. Cada fila també es pot denominar tupla o registre i a cada columna també se li pot dir camp o atribut.

Per manipular la informació utilitzem un llenguatge relacional, actualment es compta amb dos llenguatges formals l'Àlgebra relacional i el Càlcul relacional. L'Àlgebra relacional permet descriure la forma de realitzar una consulta, en canvi, el Càlcul relacional només indica el que es desitja retornar.

El llenguatge més comú per construir les consultes a bases de dades relacionals és SQL, Structured Query Language o Llenguatge Estructurat de Consultes, un estàndard implementat pels principals motors o sistemes de gestió de bases de dades relacionals.

4.2.3 Esquema

Un esquema és la definició d'una estructura (generalment relacions o taules d'una base de dades), és a dir, determina la identitat de la relació i que tipus d'informació podrà ser emmagatzemada dins d'ella, en altres paraules, l'esquema són els meta dades de la relació. Tot esquema constarà de:

- Nom de la relació (el seu identificador).
- Nom dels atributs (o camps) de la relació i els seus dominis; el domini d'un atribut o camp defineix els valors permesos per al mateix, és equivalent al tipus de dada per exemple *character*, *integer*, *string*, etc.

4.2.4 Instàncies

Una instància de manera formal és l'aplicació d'un esquema a un conjunt finit de dades. En paraules no tan tècniques, es pot definir com el contingut d'una taula en un moment donat, però també és valgut referir-nos a una instància quan

treballem o volem mostrar únicament un subconjunt de la informació continguda en una relació o taula, com per exemple:

- Certs caràcters i nombres (una sola columna d'una sola fila).
- Algunes o totes les files amb totes o algunes columnes
 - Cada fila és una tupla. El nombre de files és anomenat *cardinalitat*.
 - El nombre de columnes és anomenat *aritat o grau*.

4.3 Base de dades relacional

Una base de dades relacional és un conjunt d'una o més taules estructurades en registres (línies) i camps (columnes), que es vinculen entre si per un camp en comú, en ambdós casos posseeix les mateixes característiques com per exemple el nom de camp, tipus i longitud; a aquest camp generalment se li denomina ANEU, identificador o clau. A aquesta manera de construir bases de dades se li denomina model relacional.

Estrictament parlant, el terme es refereix a una col·lecció específica de dades però sovint s'usa, en forma errònia com a sinònim del programari usat per gestionar aquesta col·lecció de dades. Aquest programari es coneix com SGBD (sistema gestor de base de dades) relacional o RDBMS (de l'anglès relational database management system).

Les bases de dades relacionals passen per un procés al que es coneix com a normalització d'una base de dades, el qual és entès com el procés necessari perquè una base de dades sigui utilitzada de manera òptima.

Entre els avantatges d'aquest model estan:

- Garanteix eines per evitar la duplicitat de registres, a través de camps claus o claus.
- Garanteix la integritat referencial: Així en eliminar un registre elimina tots els registres relacionats depenents.
- Afavoreix la normalització per ser més comprensible i aplicable.

4.4 Disseny de la base de dades

La base de dades, doncs, serà l'encarregada de emmagatzemar tota la informació dels objectes i dels usuaris.

Al nostre estudi de viabilitat em indicat que la seguretat era un dels factors més importants de la nostra base de dades. Tenir un sistema virtualitzat i una base de dades MySQL, ens garanteix que tota la informació està dins la nostra base de dades, que hi ha restriccions de seguretat amb un usuari administrador i una contrasenya que ens permetrà l'accés a la gestió de la base de dades només pels usuaris autenticats.

A part de la seguretat dins la base de dades, hem de tenir en compte que dins la base de dades hi ha tota la informació dels usuaris, com el nom d'usuari i la seva contrasenya. Per això, em codificat la paraula de pas amb M5 de tal manera que no es pugui saber la paraula de pas dels usuaris, ni tan sols per l'administrador del sistema, sent la paraula de pas personal.

En el nostre mòdul, pel cas d'ús d'inici de sessió, una vegada l'usuari fa login, es fa una consulta a la base de dades per tal de confirmar-les. Es procedeix a realitzar una cerca a la taula on estan les dades dels usuaris, per confirmar que l'usuari i el seu password existeixen. Si no es troba, es tornarà a la pàgina inicial i no se li deixarà el pas al mòdul de modelatge d'objectes.

4.5 Descripció de les taules

A continuació es mostrarà una breu explicació de les diferents taules que fan la base de dades del mòdul de modelatge d'objectes.

4.5.1 Usuaris

A la taula usuaris s'emmagatzemen les dades dels usuaris que poden accedir al mòdul de modelatge d'objectes.

- **ID:** la id de l'usuari s'incrementarà automàticament, de tal manera que cada usuari tindrà el seu ID.
- **Nom:** indicarà el nom de l'usuari de màxim 15 caràcters.
- **Password:** emmagatzema la paraula de pas codificada amb M5
- **Administrador:** indica si l'usuari és administrador o no, si és administrador el valor serà 1, en cas de ser un usuari normal serà 0.

	Campo	Tipo	Cotejamiento	Atributos	Nulo	Predeterminado	Extra
<input type="checkbox"/>	<u>ID</u>	int(11)			No		auto_increment
<input type="checkbox"/>	Nom	varchar(15)	utf8_unicode_ci		Sí	NULL	
<input type="checkbox"/>	Paraula	varchar(40)	utf8_unicode_ci		Sí	NULL	
<input type="checkbox"/>	Administrador	tinyint(1)			No	0	

4.5.2 Objectes

A la taula objectes, es guardarà informació sobre els objectes de cada usuari, amb una relació jeràrquica pare-fill.

- **ID_objecte:** conté l'identificador únic de l'objecte, és a dir la matrícula única universal de l'objecte per tal de garantir que no hi haurà un altre objecte amb la mateixa ID.
- **ID_pare:** com és una estructura jeràrquica pare-fill, cada objecte contindrà l'ID del seu pare, com es la mateixa matrícula que ID_objecte, ens dona garanties de que cada objecte únicament pot tenir un pare.
- **Nom:** nom de l'objecte.
- **Descripció:** descripció informativa de l'objecte.
- **Categoria:** al camp categoria indiquem si és un equip o una carpeta, en cas de ser un equip el valor serà 1, si es una carpeta el valor és 2 i si és l'objecte inici de l'usuari, el valor serà 0.
- **Propietari:** indica l'ID del propietari, en cas de ser un objecte públic el propietari serà 0.
- **Publicat:** és un boolea, indica 1 si s'ha publicat i 0 si no s'ha publicat.

- **Versió:** indica la versió de l'objecte. Per definició la versió inicial d'un objecte és 0.1, cada vegada que anem modificant l'objecte la versió s'incrementa en intervals de 0.1. Quan un objecte es publica la versió passa al seu enter superior immediat, és a dir en cas de tenir un objecte a la versió 0.3, aquest quan es publiqui serà la versió 1.0. Si aquest es copia a l'usuari de nou, passarà a la versió 1.1.
- **Imatge:** conté la URL de la imatge de l'objecte.
- **Nom_individual:** conte el nom individual del nostre objecte.







	Campo	Tipo	Cotejamiento	Atributos	Nulo	Predeterminado
<input type="checkbox"/>	<u>ID_objecte</u>	varchar(50)	utf8_unicode_ci		No	
<input type="checkbox"/>	ID_pare	varchar(50)	utf8_unicode_ci		Sí	NULL
<input type="checkbox"/>	Nom	varchar(15)	utf8_unicode_ci		Sí	NULL
<input type="checkbox"/>	Descripció	varchar(40)	utf8_unicode_ci		Sí	NULL
<input type="checkbox"/>	Categoria	int(8)			Sí	NULL
<input type="checkbox"/>	Propietari	varchar(15)	utf8_unicode_ci		Sí	NULL
<input type="checkbox"/>	Publicat	tinyint(1)			Sí	NULL
<input type="checkbox"/>	Versio	decimal(3,1)			Sí	NULL
<input type="checkbox"/>	Imatge	varchar(35)	utf8_unicode_ci		Sí	NULL
<input type="checkbox"/>	Nom_individual	varchar(15)	utf8_unicode_ci		Sí	NULL

4.5.3 Categoria

Conté les categories dels objectes.

	Campo	Tipo	Cotejamiento
<input type="checkbox"/>	<u>ID_categoria</u>	int(11)	
<input type="checkbox"/>	categoria	varchar(15)	utf8_unicode_ci

De moment aquesta taula només contindrà 3 tipus d'objectes, explicats anteriorment a la taula objectes. Així que tindrem predefinits, les diferents categories.

←T→			ID_categoria	categoria
<input type="checkbox"/>			1	Equip
<input type="checkbox"/>			2	Carpeta
<input type="checkbox"/>			0	Usuari

4.5.4 Característiques

Aquí tindrem les característiques que podem posar als objectes.

- **ID_característica:** conte l'identificador únic de la característica, es un identificador incremental.
- **ID_objecte:** conté la matrícula única universal de l'objecte al qual pertany la característica.
- **Característica:** emmagatzema el nom de la característica que volem assignar a l'objecte.
- **Obligatori:** per als objectes, aquest cap indica si la característica que posem és obligatòria per al sistema, de tal manera que serà restrictiva en el nostre sistema, no podent no complir les característiques que ha de tenir l'objecte.
- **Valor:** valor de la característica si necessitem unitats.
- **Descripció:** camp de text per posar una descripció extensa de la nostra característica, per poder aclarir dubtes o posar comentaris.

	Campo	Tipo	Cotejamiento
<input type="checkbox"/>	<u>ID_característica</u>	varchar(50)	utf8_unicode_ci
<input type="checkbox"/>	ID_objecte	varchar(50)	utf8_unicode_ci
<input type="checkbox"/>	característica	varchar(40)	utf8_unicode_ci
<input type="checkbox"/>	obligatori	tinyint(1)	
<input type="checkbox"/>	valor	varchar(10)	utf8_unicode_ci
<input type="checkbox"/>	descripcio	varchar(150)	utf8_unicode_ci

4.5.5 Imatges

Conté les URL de les imatges que poden caracteritzar els nostres objectes.

- **ID_imatge:** identificador únic de les imatges.
- **URL:** contindrà la URL de la nostra imatge dins el nostre sistema, de tal manera que el nostre codi HTML ho pugui interpretar i trobi la ruta dins el nostre sistema per garantir que podrà mostrar la imatge.

	Campo	Tipo	Cotejamiento	Atributos	Nulo	Predeterminado	Extra
<input type="checkbox"/>	<u>ID_imatge</u>	int(11)			No		auto_increment
<input type="checkbox"/>	URL	varchar(100)	utf8_unicode_ci		No		

4.5.6 Diccionari

Conté els diferents diccionaris de senyals dels objectes.

- **Codi diccionari:** tindrà el ID del diccionari.
- **Nom diccionari:** contindrà el nom dels diccionaris emmagatzemats.

	Campo	Tipo	Cotejamiento
<input type="checkbox"/>	<u>Codi diccionari</u>	varchar(15)	utf8_unicode_ci
<input type="checkbox"/>	nom_diccionari	varchar(30)	utf8_unicode_ci

4.5.7 Senyals diccionari

Té el contingut de les senyals dels diccionaris.

- **Codi_senyal:** conté l'identificador únic del senyal.
- **Codi_diccionari:** indica quin és el diccionari al que pertany el senyal.
- **Tipus_senyal:** indica el tipus de senyal.

- **Criticitat:** és un camp numeral, que indica amb una escala el nivell de criticitat del nostre senyal (0 poc crític, 10 molt crític).
- **TAG:** emmagatzema el tag del senyal.
- **Funció:** pot emmagatzemar breument la funció del senyal.
- **Descripció:** ens permet descriure més extensament el nostre senyal.

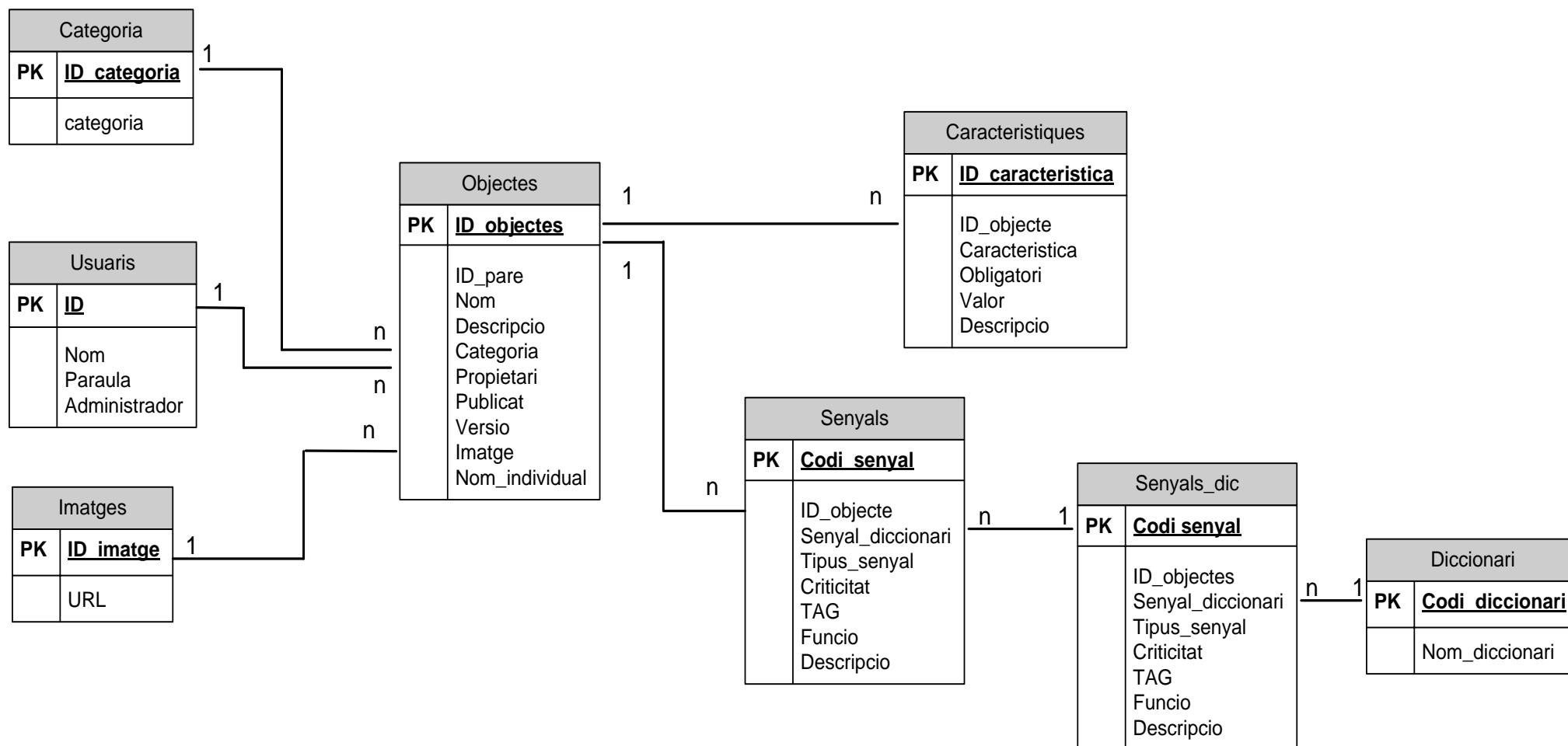
	Campo	Tipo	Cotejamiento
<input type="checkbox"/>	<u>codi_senyal</u>	varchar(30)	utf8_unicode_ci
<input type="checkbox"/>	codi_diccionari	varchar(15)	utf8_unicode_ci
<input type="checkbox"/>	tipus_senyal	varchar(3)	utf8_unicode_ci
<input type="checkbox"/>	criticitat	int(3)	
<input type="checkbox"/>	TAG	varchar(10)	utf8_unicode_ci
<input type="checkbox"/>	funcio	varchar(15)	utf8_unicode_ci
<input type="checkbox"/>	descripcio	varchar(25)	utf8_unicode_ci

4.5.8 Senyals

Conté els senyals que em assignat als nostres objectes, té els mateixos camps que senyals del diccionari, però no conté l'identificador de diccionari, sinó la matrícula única de l'objecte al qual pertany.

	Campo	Tipo	Cotejamiento
<input type="checkbox"/>	<u>Codi_senyal</u>	int(30)	
<input type="checkbox"/>	ID_objecte	varchar(50)	utf8_unicode_ci
<input type="checkbox"/>	Senyal_diccionari	varchar(30)	utf8_unicode_ci
<input type="checkbox"/>	tipus_senyal	varchar(3)	utf8_unicode_ci
<input type="checkbox"/>	criticitat	int(3)	
<input type="checkbox"/>	TAG	varchar(10)	utf8_unicode_ci
<input type="checkbox"/>	funcio	varchar(15)	utf8_unicode_ci
<input type="checkbox"/>	descripcio	varchar(60)	utf8_unicode_ci

4.5.9 Diagrama model de dades



4.6 Interfícies

A continuació mostrarem les diferents interfícies. Les diferències entre la interfície de l'administrador i d'un usuari normal és la capacitat de poder crear o eliminar usuaris, pel referent a l'entorn operatiu del mòdul d'objectes, és igual tant per l'usuari com per l'administrador.

4.6.1 Pàgina de login



giscmo, mòdul d'objectes

presentació

introducció

Fet per Marc Àngel, GISC MO Mòdul d'objectes 2009-2010

Que és GISC MO?

GISC MO pretén ser una eina software per donar suport al disseny d'entorns de control industrial SCADA. Actualment no es pot trobar, en el mercat, un software GNU que sigui capaç de fer el modelatge de sistemes de control, creació de codi automatitzat per diferents sistemes de control, etc. GISC MO serà un projecte Open Source que aportarà una solució a aquest tipus de models.

A més a més, s'utilitzarà un entorn web dinàmic que serà modificable per qualsevol programador a conveni de les seves necessitats.

El projecte consta en un mòdul de GISC MO, concretament el mòdul de modelatge d'objectes.

El mòdul de modelatge d'objectes, servirà per modelar els elements que utilitzarem posteriorment per fer el disseny dels nostres projecte de control industrial.

Des de el punt de vista de la plataforma GISC MO un objecte es una sèrie d'informacions, que especifiquen tan dades tècniques com comercial d'un equip. Aquests equips han de esser entesos com model de elements del mon real com poden ser motors, vàlvules, llums automàtics, etc. Aquest elements s'agruparan en llibreries que seran d'us i consulta publica.

Què vol dir, poder modelar objectes? S'ha de poder dissenyar un objecte des de 0, això implicarà poder crear una estructura jeràrquica amb un nombre indeterminat de nivells, que en conjunt modelin un element del mon real. S'ha de poder associar una imatge que caracteritzi cada element de la jerarquia. A més, se li han de poder donar uns identificadors únics, una sèrie de característiques i uns senyals les quals estaran definides anteriorment, dins els corresponents diccionaris establerts per el mòdul de gestió de diccionaris.

Tots els objectes definits mitjança'ns el nostre aplicatiu, estaran gestionats per personal tècnic qualificat que serà l'encarregar i responsable de la creació, classificació i publicació de les llibreries de treball.

login

Usuari

Password

Login

menú

Introducció

©GISC MO - Mòdul d'objectes - Projecte 2009-2010 per Marc Àngel Rodríguez

Des de la pàgina de login, vàlida per qualsevol usuari estigui creat o sigui invitat, es pot veure una petita introducció a que és GISCMO.

A la dreta tenim el menú de Login, a través del qual els usuaris poden accedir amb el seu nom o contrasenya.

4.6.2 Interfície administrador

Quan fem login amb un usuari administrador , tenim un menú una mica diferent del d'un usuari normal. Un administrador tindrà la opció de poder accedir a la gestió d'usuaris del mòdul d'objectes o bé anar a GISCMO - Mòdul d'objectes.



4.6.3 Gestió d'usuaris

Des de la gestió d'usuaris podem crear un nou usuari o eliminar els usuaris existents.



4.6.3.1 Afegir nou usuari:

The screenshot shows a web application titled 'menú'. Below the title is a subtitle: 'Fet per Marc Àngel, GISCMO Mòdul d'objectes 2009-2010'. The main content area has a dark background. At the top, there are two buttons: 'Afegir nou usuari' (highlighted with a red dashed border) and 'Eliminar usuari'. Below these buttons, there are three input fields: 'Nom usuari:' followed by a text box, 'Contrasenya' followed by a text box, and 'Administrador' followed by a checkbox. Below the input fields are two buttons: 'Afegir usuari' and 'Cancel·la'. At the bottom of the form area, there are two buttons: 'Torna enrerre' and 'Tanca sessió'.

Té 2 camps, un per el nom d'usuari i un camp per la paraula de pas, també té un checkbox per si volem donar-li permisos d'administrador a l'usuari.

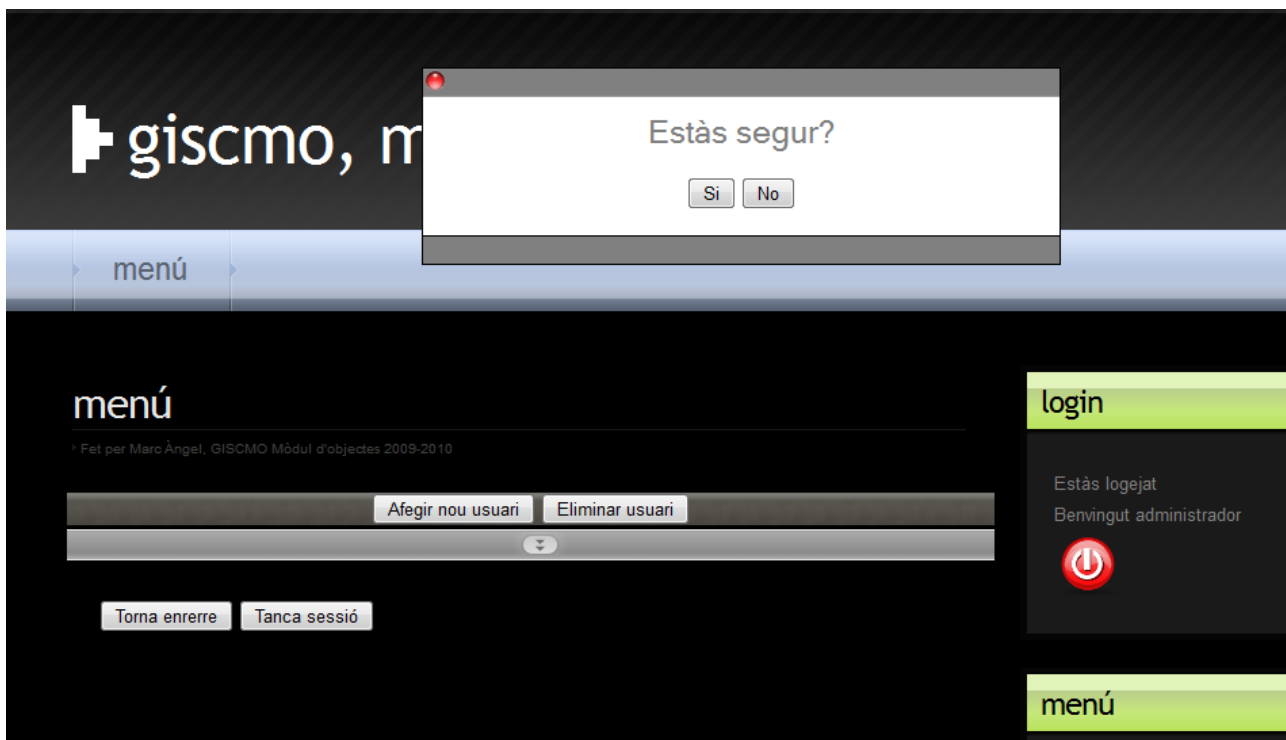
4.6.3.2 Eliminar usuari:

The screenshot shows the same web application 'menú' with the same subtitle. In this view, the 'Eliminar usuari' button is highlighted with a red dashed border. Below the buttons, there is a table with two columns: 'Usuari' and 'Selecciona'. The table contains three rows of data:

Usuari	Selecciona
marc	<input checked="" type="checkbox"/>
administrador	<input checked="" type="checkbox"/>
eladio	<input checked="" type="checkbox"/>

At the bottom of the form area, there are two buttons: 'Torna enrerre' and 'Tanca sessió'.

Seleccionem l'usuari que volem, una vegada seleccionat l'usuari ens demana confirmació.



També tenim sempre un botó per tancar sessió des de qualsevol dels menús o bé sota el menú principal, també tenim un botó per tornar a la pàgina anterior.

4.6.4 Interfície usuari

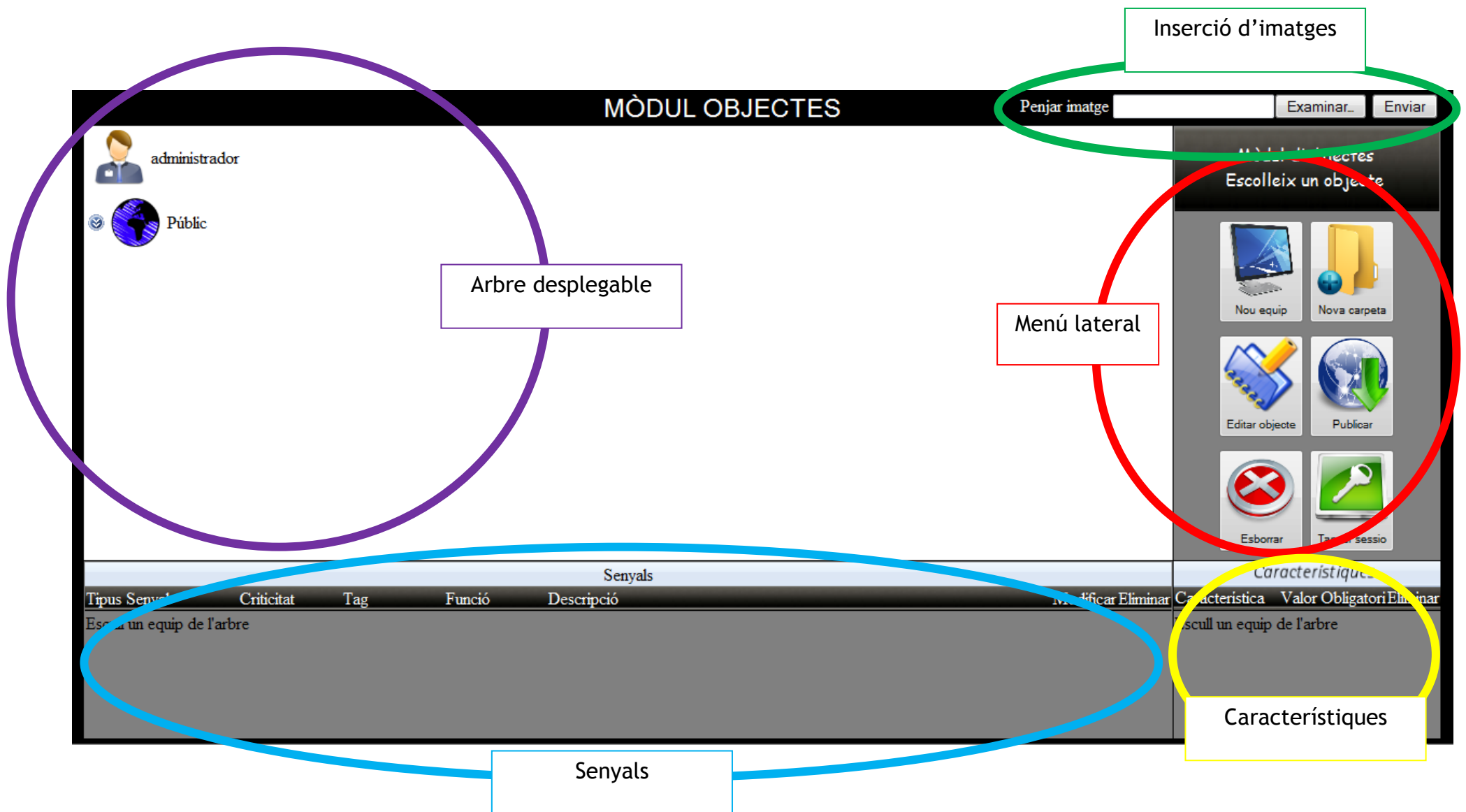


El menú d'usuari únicament té una pantalla de benvinguda amb un accés al mòdul d'objectes o un botó de “logout”.

4.6.5 Interfície mòdul modelatge d'objectes GISCMO

El mòdul de modelatge d'objectes és igual tant per un usuari normal com per un administrador. Aquesta interfície és una interfície dinàmica amb la possibilitat de mostrar finestres per les diferents funcions. Té un arbre a la part esquerra, un menú lateral a la part dreta i a les parts inferiors, els senyals i les característiques.

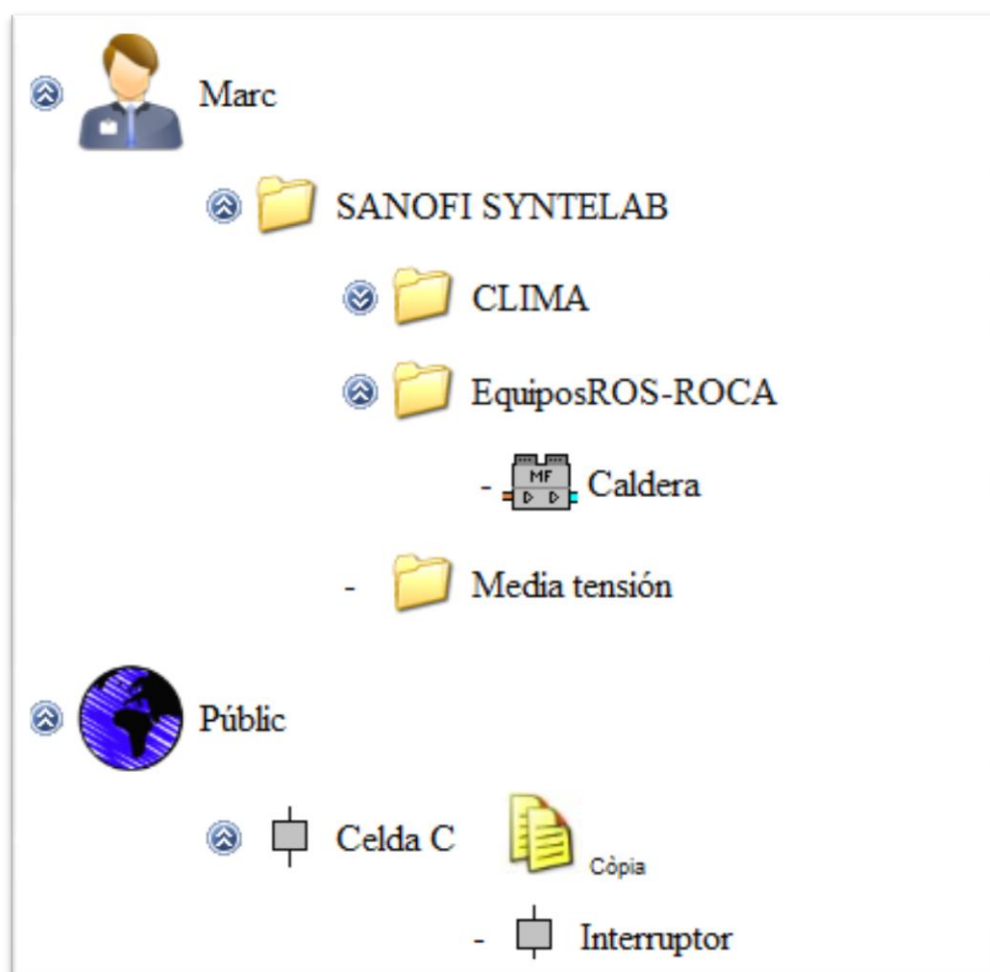
Podem fer una selecció d'un objecte del nostre usuari, o bé una selecció d'un objecte publicat, poden fer diferents accions per cada un d'aquests objectes.



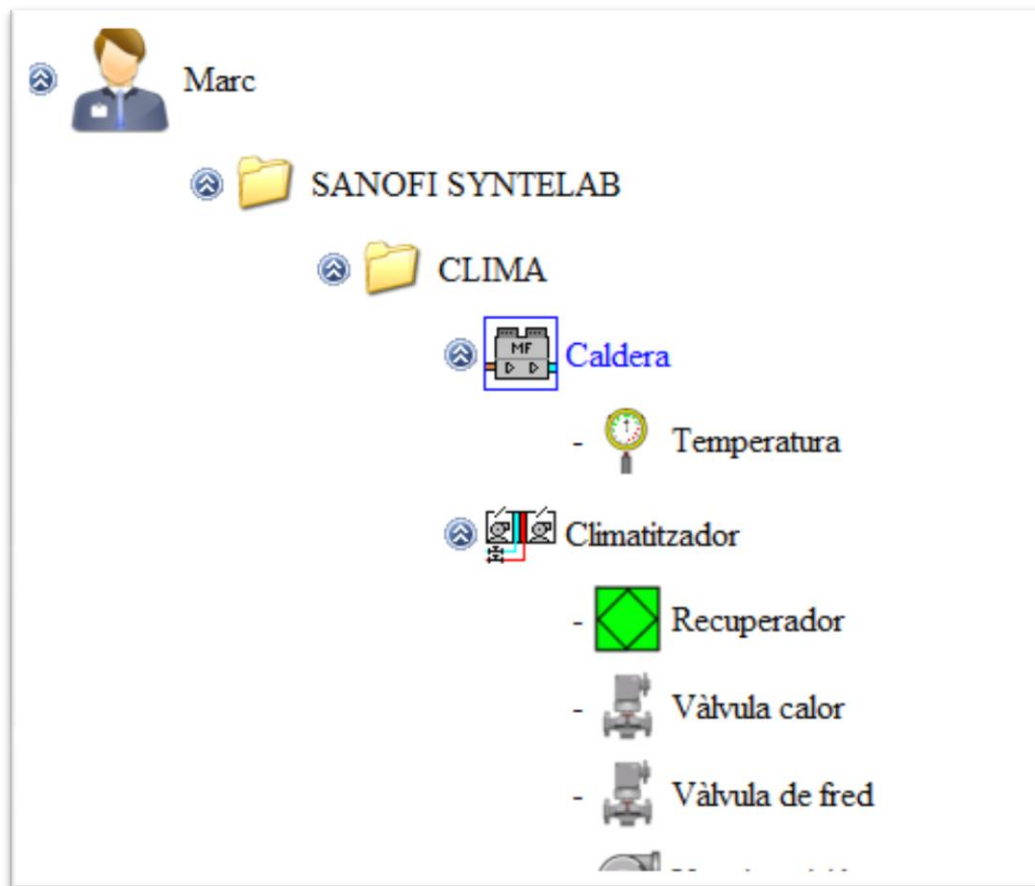
4.6.5.1 Arbre desplegable

En l'arbre desplegable tenim, els objectes de l'usuari per una part, i els objectes publicats per una altra part. Podem veure un exemple d'un usuari anomenat Marc, els seus objectes i els objectes que hi ha publicats.

Com veïem l'usuari pot organitzar tots els seus objectes en carpetes. No hi ha límit de fills, però, només podem crear carpetes sota una altra carpeta o sota l'usuari.

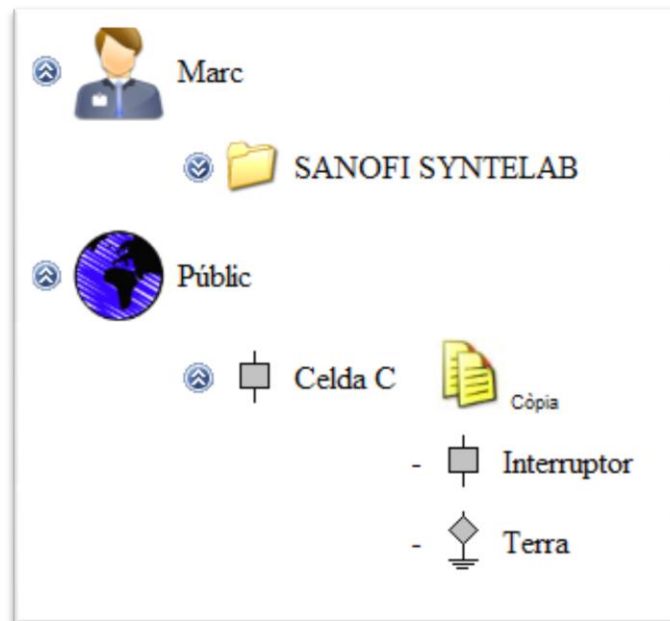


De l'arbre podem seleccionar un objecte qualsevol, de tal manera que els altres panells s'actualitzen automàticament amb les dades del objecte seleccionat.



L'objecte que volem seleccionar es marca amb un quadrat blau, el nom també es ressalta en blau.

A més, els objectes publicats es poden copiar de nou com a versió superior per fer modificacions o utilitzar-ho per un nou objecte. Per fer-ho, tenim un botó de copiar al costat de l'objecte.

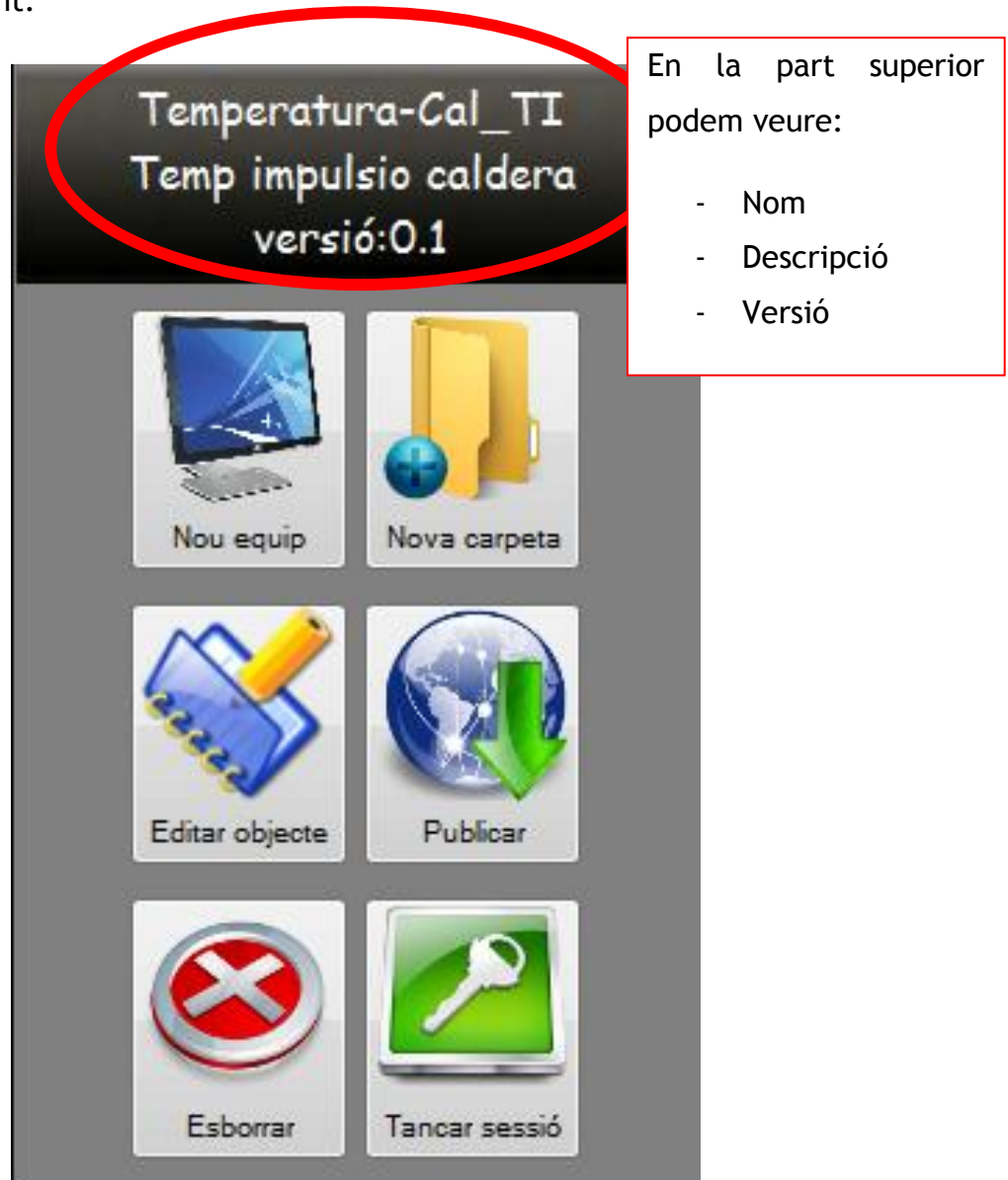


4.6.5.2 Menú lateral

Una vegada seleccionat un objecte, es recarrega el menú lateral, on tenim:

- Informació del nom, descripció de l'objecte i versió actual de l'objecte.
- Botó Nou equip: ens permet crear un equip nou per sota de l'equip seleccionat fent un fill per aquest. L'objecte seleccionat pot ser un equip, una carpeta o l'usuari, en cas de ser l'usuari, vol dir que volem crear un objecte nou a l'arrel del nostre usuari.
- Botó nova carpeta: ens permet crear una nova carpeta sota l'usuari o sota una altra carpeta, en cas de voler crear una carpeta sota un objecte, es notificarà amb un missatge d'error.
- Editar objecte: podrem modificar les dades de l'objecte seleccionat.
- Publicar: ens permet publicar un equip que ja esta totalment finalitzat.
- Esborrar: ens permet esborrar un equip i tots els seus fills del nostre arbre, també es poden esborrar les carpetes, si s'esborra la carpeta també s'esborrarà tota la informació que conté aquesta.
- Tancar sessió: tanca sessió del nostre usuari i torna al menú inicial.

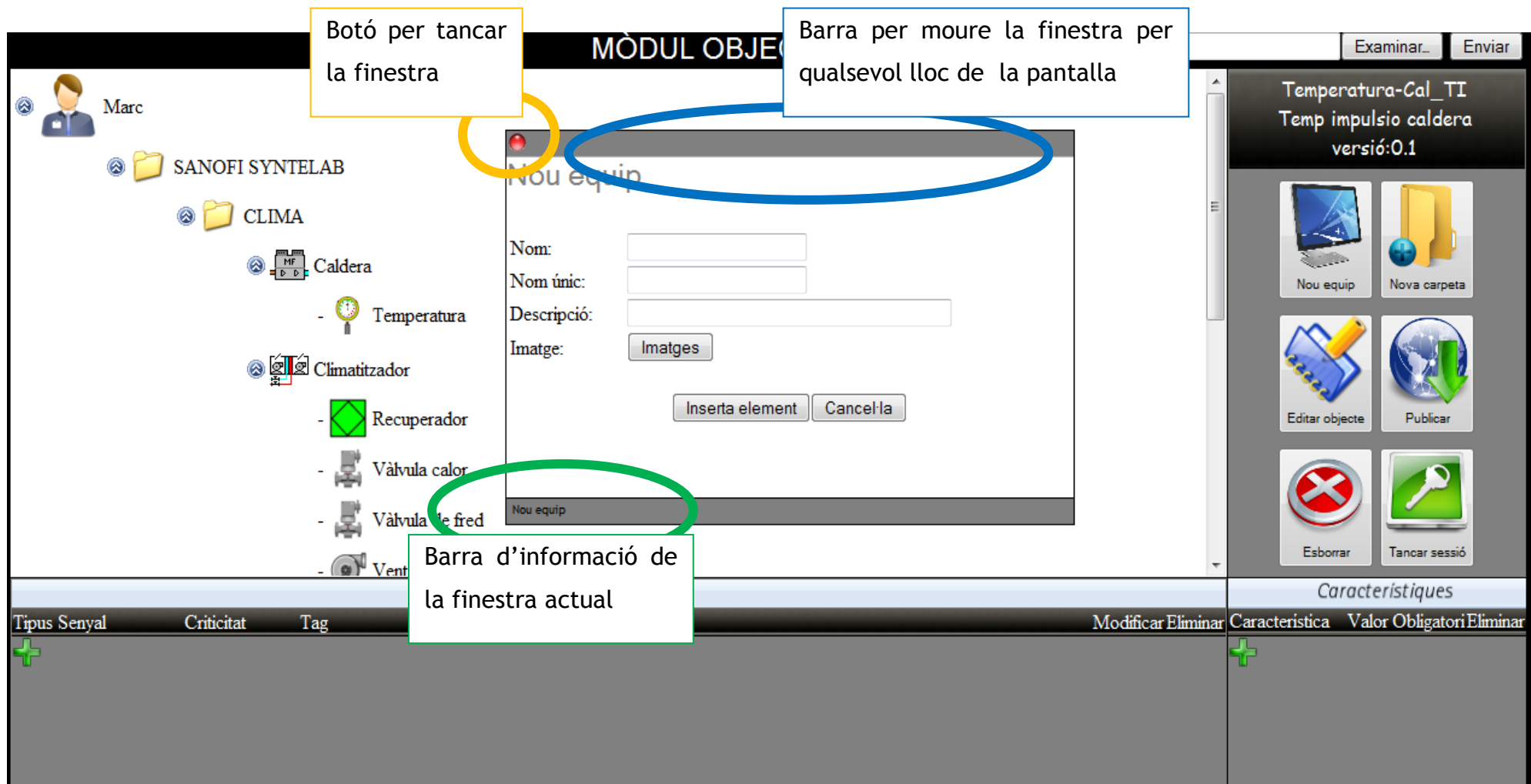
Per cada una de les diferents accions, ens apareixerà una finestra al mig de l'entorn operatiu GISCMO modelatge d'objectes que ens permetrà interactuar més còmodament.



4.6.6 Finestres menú lateral

Una vegada seleccionat un objecte des de l'arbre, podem fer qualsevol de les opcions del menú lateral. Aquest sistema de finestres també s'utilitza per les senyals i per les característiques.

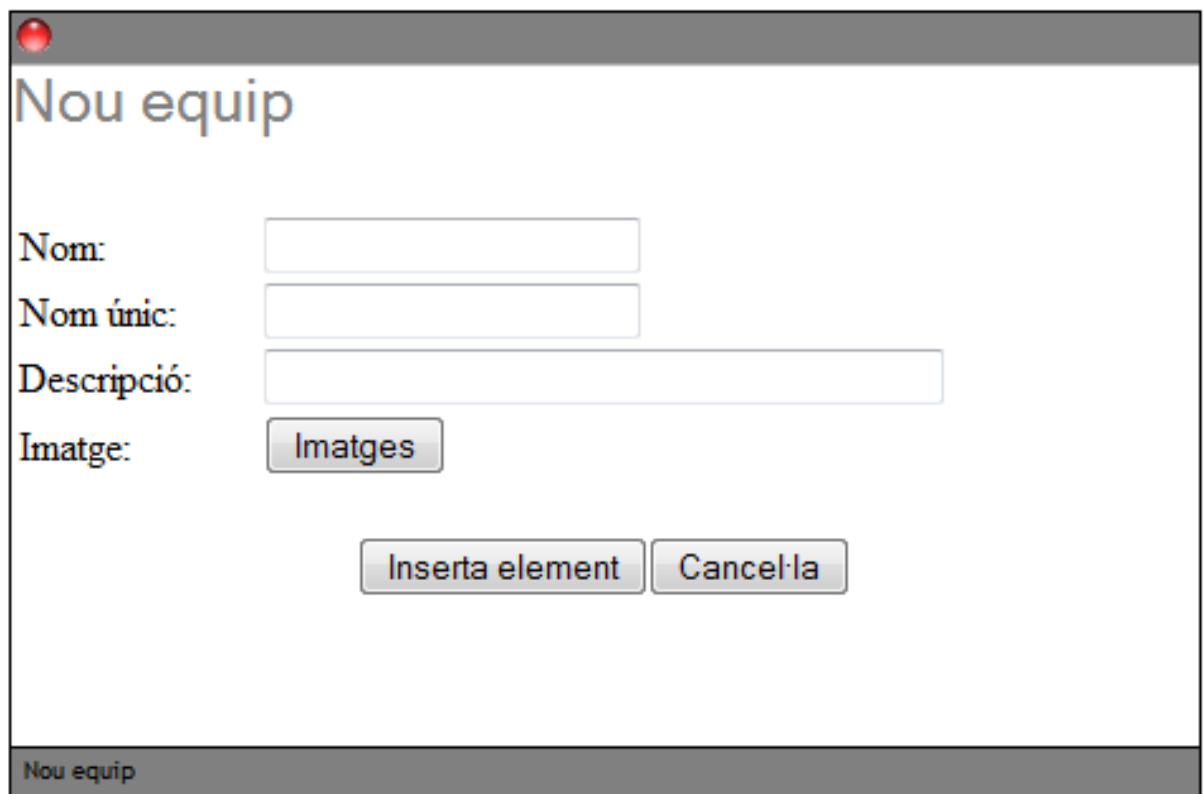
Veiem un exemple de com es mostren les finestres dins el nostre entorn:



A continuació mostrarem algunes de les accions possibles des d'aquest menú lateral.

4.6.6.1 Finestra nou equip i modificar equip

En la finestra nou equip tenim un formulari que ens permet introduir el nom, nom únic, descripció i escollir una imatge. També tenim un botó per cancel·lar i tancar la finestra i un botó per insertar l'element definitivament.



Nou equip

Nom:

Nom únic:

Descripció:

Imatge:

Nou equip

Quan donem al botó imatges, s'obre una nova finestra amb totes les imatges disponibles al servidor per identificar els objectes, únicament s'ha de fer "click" a la imatge desitjada. Una vegada seleccionada la imatge, cal pressionar "Inserta element" per afegir un nou objecte.



Modificar equip, és exactament igual, no obstant a l'hora d'obrir la finestra tindrem les dades actuals de l'objecte per si les volem modificar, una vegada canviades les dades, només s'ha de seleccionar l'opció "Modificar equip".

Modificar equip: Temperatura

Nom:

Temperatura


Nom únic:

Cal_TI

Descripció:

Temp impulsio caldera

Imatge:



Imatges

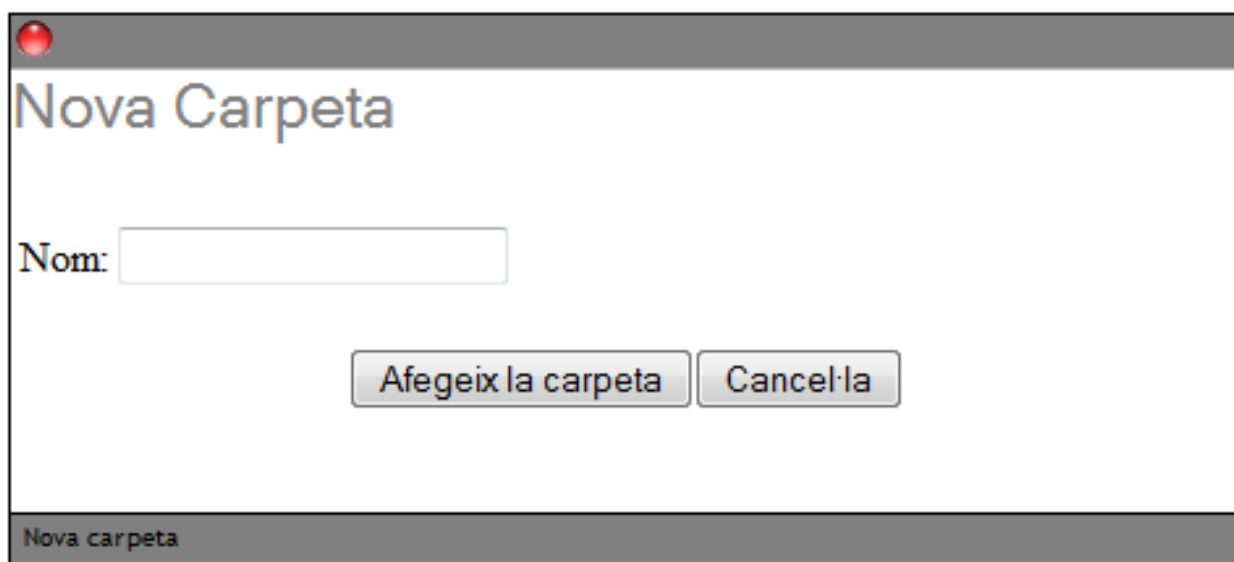
Modificar equip

Cancel·la

Editar equip

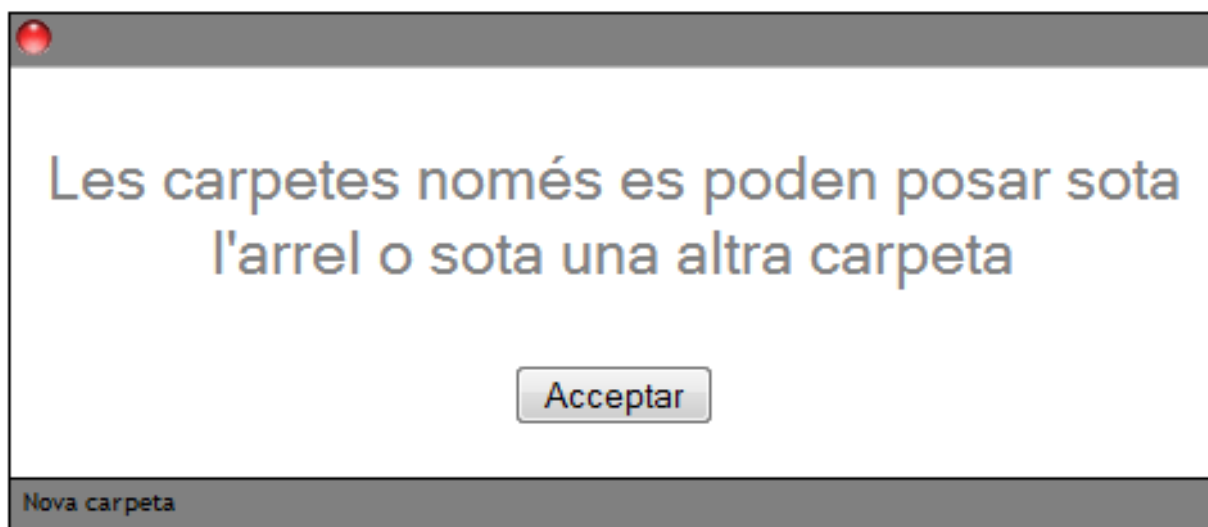
4.6.6.2 Finestra nova carpeta

Nova carpeta té el mateix funcionament que nou equip, no obstant les carpetes únicament les podem crear sota l'arrel o bé sota una altra carpeta. Es tracta d'un formulari on es demana únicament el nom de la carpeta a crear.



The image shows a dialog box titled "Nova Carpeta". It has a standard window header with a red close button. The main area contains the label "Nom:" followed by a text input field. Below the input field are two buttons: "Afegeix la carpeta" and "Cancel·la". At the bottom of the dialog, there is a status bar that says "Nova carpeta".

En cas de crear una carpeta sota un objecte o en un element publicat, ens donarà un missatge d'informació d'error.

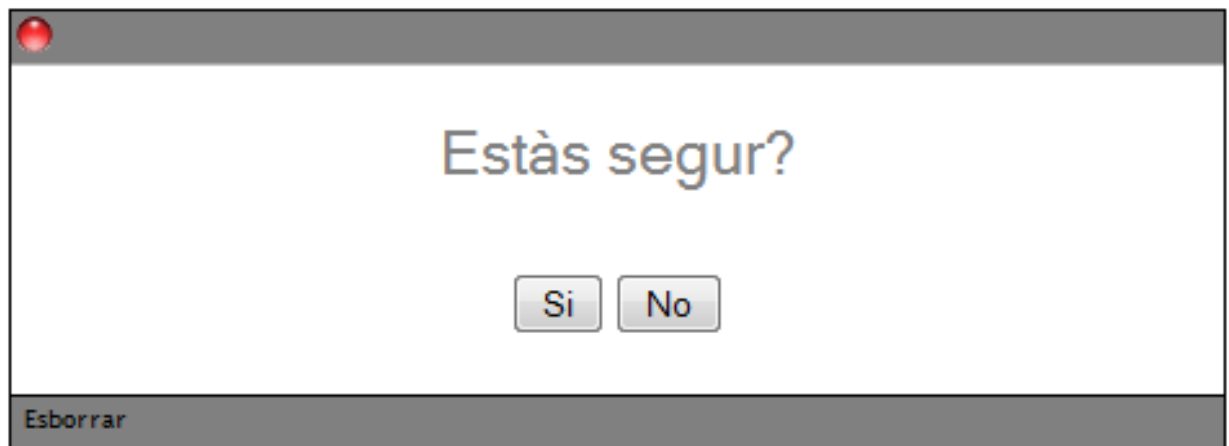


The image shows an error message dialog box. It has a standard window header with a red close button. The main area contains the text "Les carpetes només es poden posar sota l'arrel o sota una altra carpeta" in a large font. Below the text is a single button labeled "Acceptar". At the bottom of the dialog, there is a status bar that says "Nova carpeta".

4.6.6.3 *Esborrar*

El botó esborrar, permet eliminar qualsevol objecte propietat de l'usuari. No permet esborrar un objecte publicat ni tampoc, l'arrel.

Quan prenem el botó esborrar ens apareix un missatge de confirmació, per assegurar que volem fer l'acció d'esborrar.



4.6.6.4 *Publicar i tancar sessió*

L'acció Publicar, permet publicar qualsevol objecte de la nostre propietat que vulguem establir com a definitiu. Per publicar, s'haurà disposar de la conformitat d'un administrador per confirmar que les dades son correctes. Una vegada verificat l'objecte, es podrà fer public per tal que qualsevol usuari pugui fer-ne ús.

El botó tancar sessió, tancarà la sessió de l'usuari actual i tornarà a la pàgina principal donant pas a un altre usuari.

4.6.7 *Senyals*

Al menú de senyals, podem caracteritzar el nostre objecte amb els tipus de senyals disponibles als diccionaris. Podem seleccionar les senyals dintre d'una llista de diccionaris, i copiar-la per al nostre objecte.

Una vegada copiat el senyal, també la podem modificar per si necessitem fer alguna variació del senyal, donar-li un nom únic, etc.








Els senyals dels diccionaris, son senyals de caire estàndard, que assegurin el funcionament amb els objectes, no obstant, hi ha diversos camps, que es permeten modificar perquè no son vitals per el funcionament de l'objecte no variï.

De la mateixa manera que al menú lateral, al menú de senyals també tenim un sistema de finestres que ens facilita la interacció amb la interfície del mòdul de modelatge d'objectes.

També podem trobar diversos botons que ens permeten fer les accions:

- Botó afegir nou senyal: obra una finestra a on podem seleccionar els diferents diccionaris disponibles i els senyals que volem afegir al nostre objecte.
- Botó modificar: ens permet modificar alguns dels camps del senyal del nostre objecte
- Botó eliminar: ens elimina el senyal del nostre objecte, només cal prémer el botó per eliminar el senyal, aquí no ens demanarà una confirmació per eliminar.

Cal tenir present, que els senyals només es poden afegir als equips, no es podran afegir a l'arrel ni a les carpetes.

Tipus Senyal	Criticat	Tag	Funció	Descripció	Modificar	Eliminar
DO	0	PM	Comandament	Permissiu de Stop/Start		
DI	0	ALMxxx	Alarma	Alarma genérica		
PI	0	LR	Estat	Indicador local o remot		
						

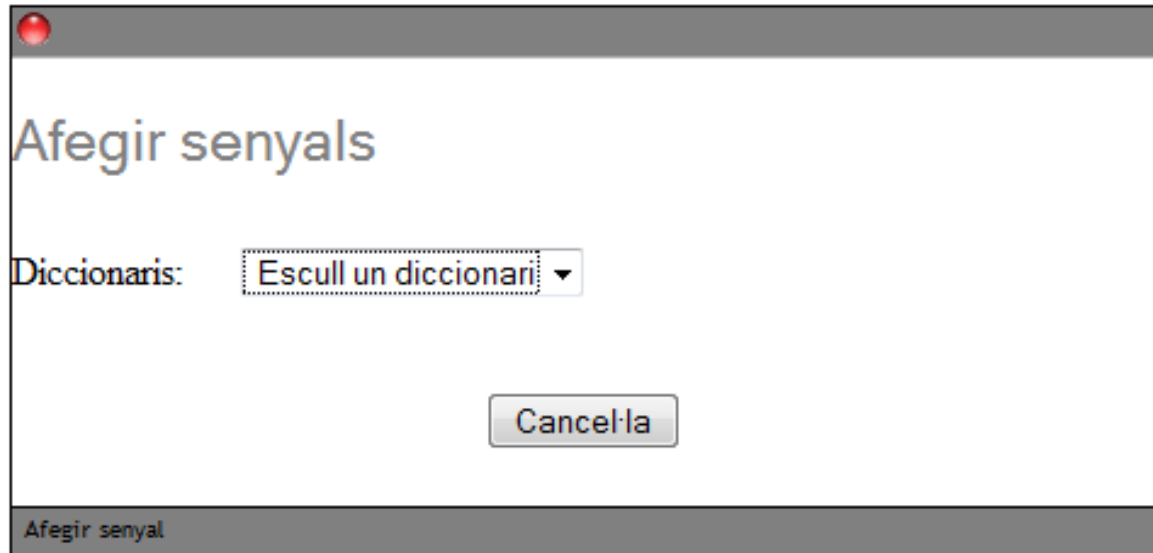
Botons eliminar

Botons afegir nou senyal

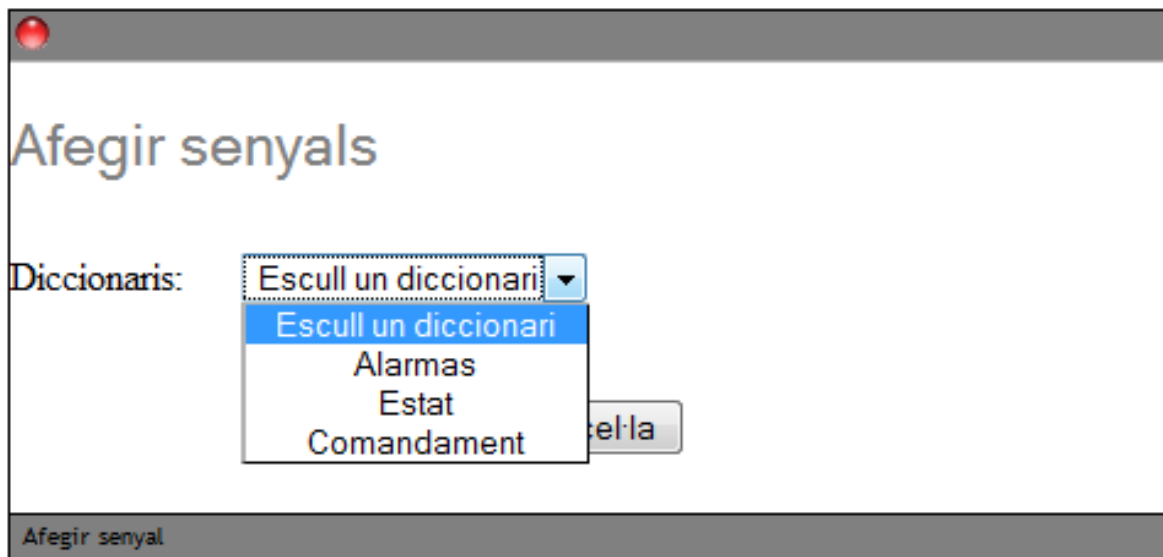
Botons modificar

4.6.7.1 Afegir senyal

Quan prenem el botó afegir senyal, ens apareixerà doncs, una finestra a on podrem seleccionar els diccionaris disponibles.



Els diccionaris es mostraran en una llista desplegable, de la qual haurem de seleccionar-ne un.



Una vegada seleccionat el diccionari, ens mostrarà automàticament tots els senyal que pertanyien al diccionari seleccionat.

Per afegir un senyal, cal prémer al botó selecciona, automàticament tornarem a la nostra llista de senyals, i veurem el senyal afegit.

Afegir senyals

Diccionari: Alarmas

Codi Senyal	Tipus Senyal	Criticitat	Tag	Funció	Descripció	Selecciona
089397845jj65	DI	4	ALMG	Alarma	General grup electrògen	
089397845jj54	DI	1	ALMTxx	Alarma	Temperatura trafo nivell	
089397845jj23	DI	4	FTA	Alarma	Falta de tensió circuits	
089397845jj452	DI	0	ALMxxx	Alarma	Alarma genèrica	
089397845jj1078	DI	0	TER	Alarma	Magnetotèrmic de alimenta	
089397845jjwef2	DI	0	MND	Alarma	Comandament no disponible	
089397845jjCASD34DI		4	Alarma	FCC	Falta CC o alimentació 11	
089397845jj321	DI	2	ALM_BAT	Alarma	Baixa tensió de bateria	
089397845jjh	DI	2	FTxxx	Alarma	Absència de tensió	

Botons
selecciona

Cancel·la

Afegir senyal

4.6.7.2 Modificar senyal

Modificar senyal és semblant a modificar equip, ens mostra una finestra amb les dades actuals del senyal, i ens deixa fer canvis en els valors dels camps, una vegada finalitzada la modificació, cal prémer “Modificar senyal”.

Una vegada acabada l’operació, el senyal es veurà actualitzat amb els canvis a la llista dels nostres senyals.

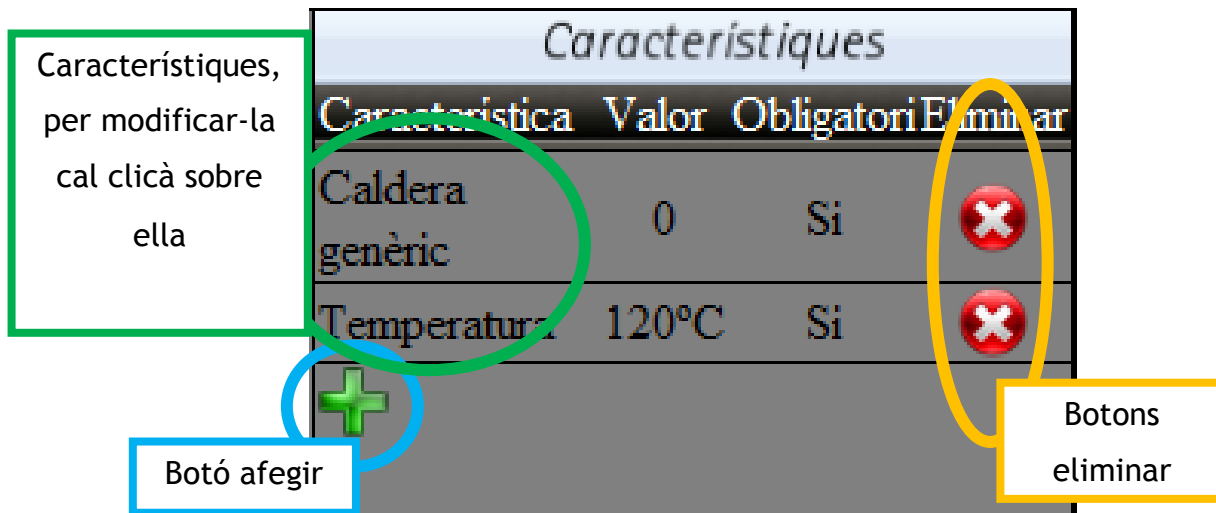


The screenshot shows a window titled "Modificar senyal: 89489572jjfna". Inside the window, there are five labeled text input fields arranged vertically: "Tipus_senyal:" with the value "DO", "Criticitat:" with the value "0", "Tag:" with the value "PM", "Funció:" with the value "Comandament", and "Descripció:" with the value "Permissiu de Stop/Start". Below these fields are two buttons: "Modifica senyal" and "Cancel·lar". At the bottom of the window, there is a status bar that says "Modificar senyal".

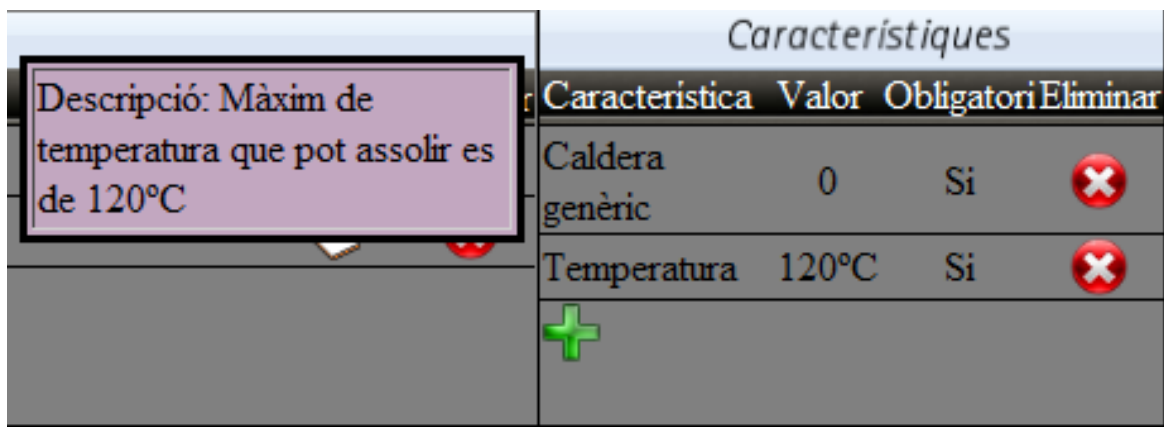
4.6.8 Característiques

El menú característiques funciona de manera semblant al menú senyals, per afegir una característica nova cal prémer el botó afegir característica, també tenim la opció de modificar la característica si la seleccionem, i tenim un botó per eliminar-la.

De la mateixa manera, funciona amb finestres perquè sigui més fàcil i intuïtiu.



No obstant, en el menú característiques, si ens posem a sobre d'una característica, apareixerà un ToolTip d'informació en el que es mostrarà la seva descripció.

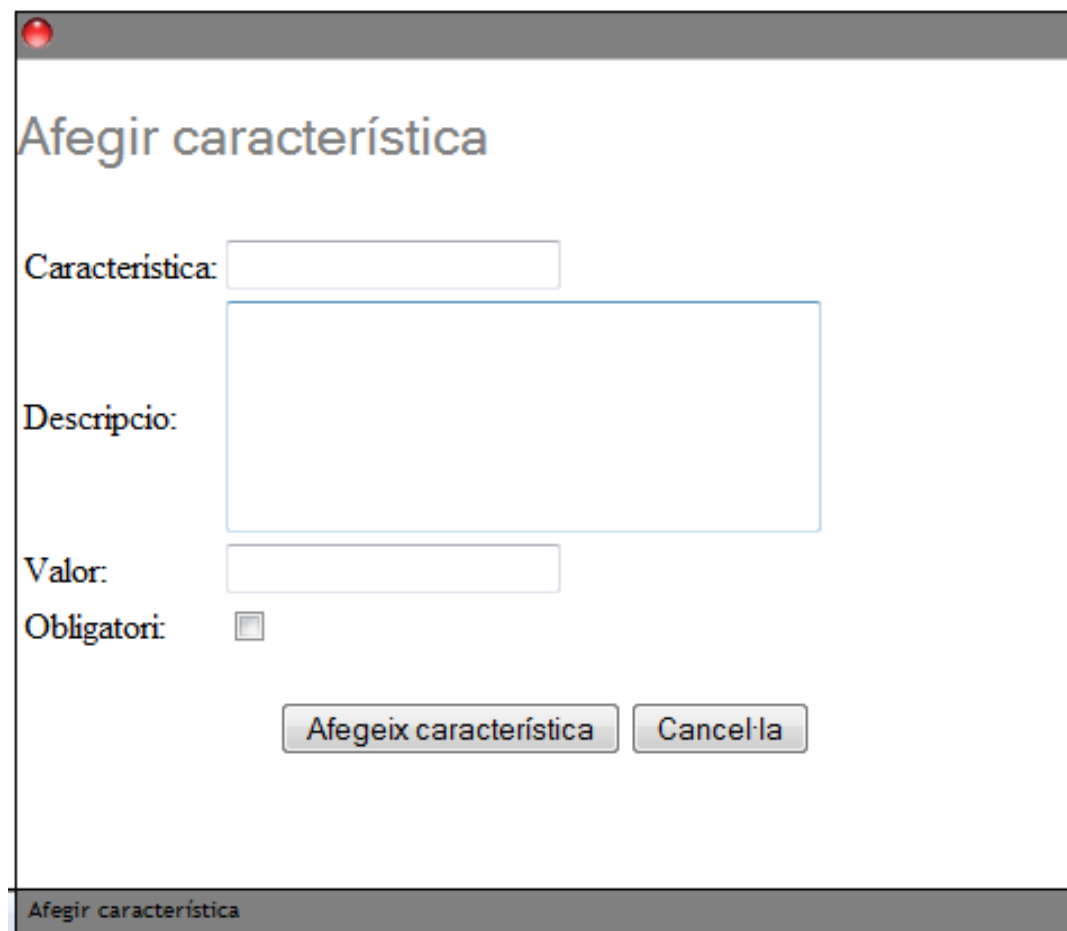


4.6.8.1 Afegir característica

Prement al botó afegir característica, apareixerà una finestra per afegir-ne de noves i tindrà els diferents camps on introduir les dades de la nova característica.

El camp Descripció, és un camp de text d'extensió màxima de 150 caràcters, també podem trobar un "checkbox" per indicar si la característica es d'us obligatori en les futures instàncies de l'objecte.

Una vegada omplerts els camps, cal donar a "Afegir característica".



Afegir característica

Característica:

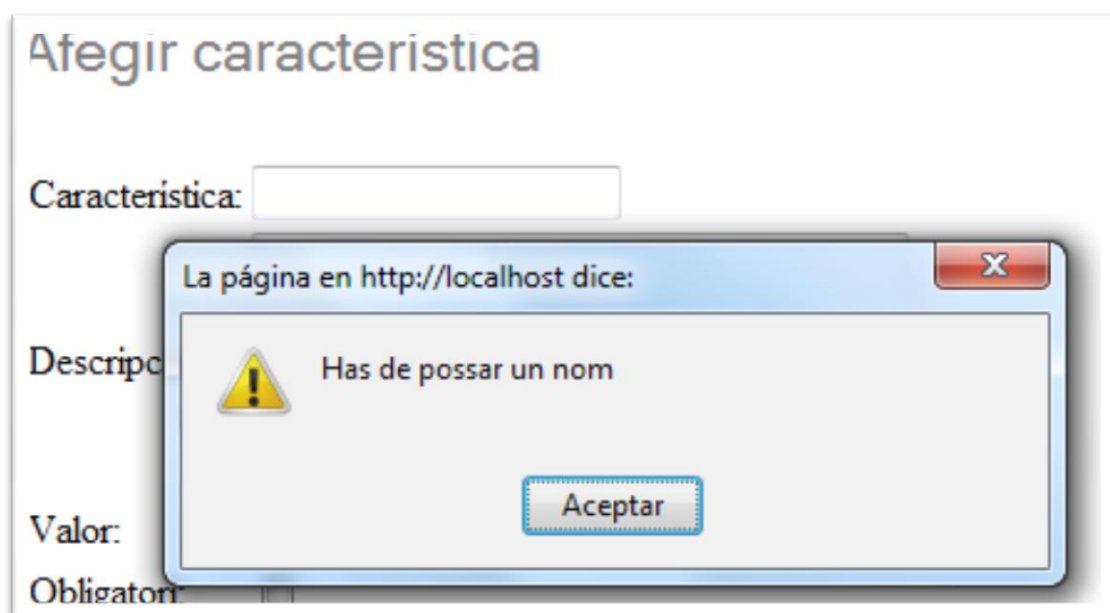
Descripció:

Valor:

Obligatori: ☐

Afegir característica

En cas de que es deixin camps en blanc, s'indica en quin camp no s'ha posat informació amb un missatge d'alerta.



Afegir característica


Característica:

Descripció:

Valor:

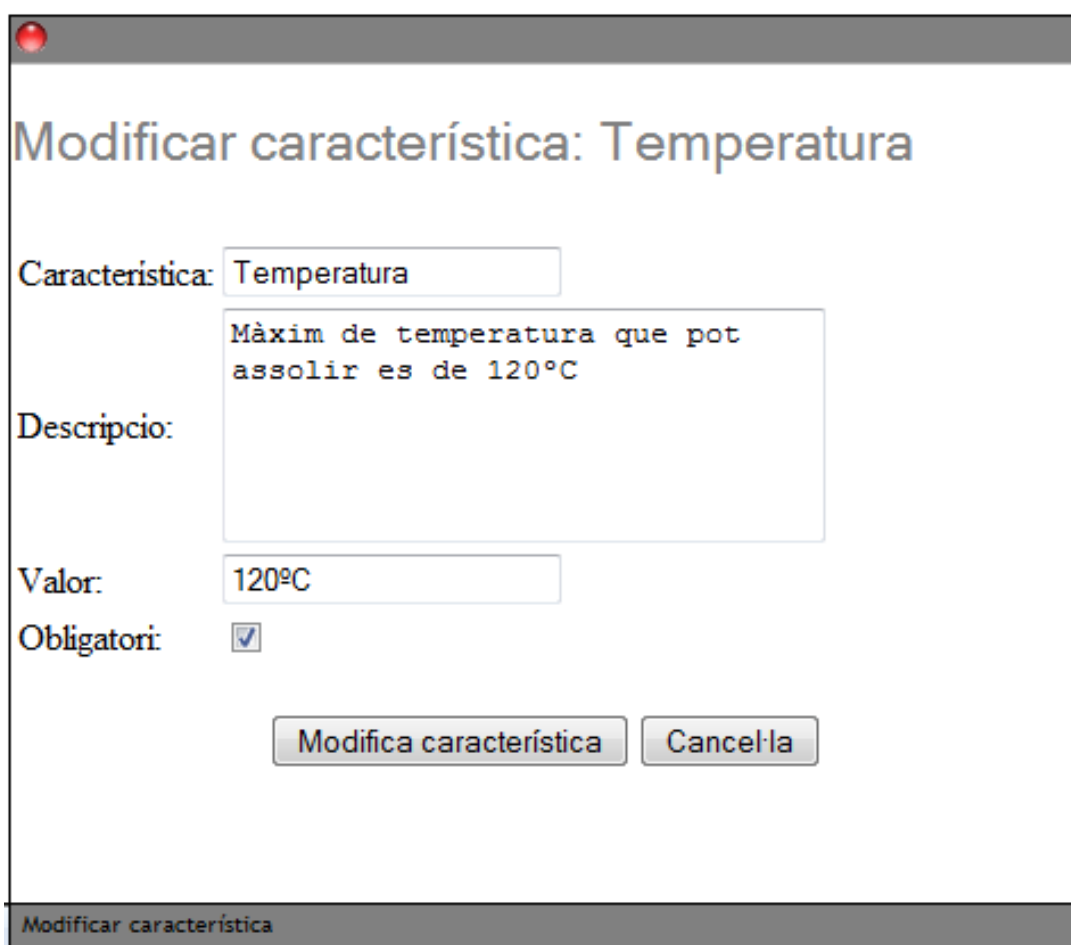
Obligatori: ☐

La pàgina en <http://localhost> dice:

 Has de posar un nom

4.6.8.2 *Modificar característica*

Pressionant sobre una característica, s'obrirà la finestra “modificar característica” que funciona de la mateixa manera que modificar senyal i modificar equip. Ens mostrà els valor dels diferents camps de la característica i ens permet modificar el seu valor. Una vegada modificat el valor veurem el valor actualitzat a la llista de característiques.



The image shows a software window titled "Modificar característica: Temperatura". It contains the following fields and controls:

- Característica:** A text box containing the word "Temperatura".
- Descripció:** A text area containing the text "Màxim de temperatura que pot assolir es de 120°C".
- Valor:** A text box containing "120°C".
- Obligatori:** A checkbox that is checked (indicated by a blue checkmark).
- Buttons:** Two buttons at the bottom: "Modifica característica" and "Cancel·la".
- Footer:** A grey bar at the bottom of the window with the text "Modificar característica".

5 Implementació

5.1 Introducció

En aquest capítol detallarem les tècniques que s'han utilitzat per dur a terme la implementació del mòdul de modelatge d'objectes de GISCMO

Després d'estudiar els requeriments en l'anàlisi i veure com poder complir els requeriments en el disseny, ara utilitzarem totes les eines seleccionades (HTML, PHP, Javascript, CSS, Ajax) per poder dur a terme el desenvolupament del mòdul.

Ja que l'objectiu és explicar com s'ha implementat el mòdul de modelatge d'objectes, es comentarà codi desenvolupat, per això faré una explicació de les parts més importants de manera que sigui fàcil d'entendre i es vegi la utilitat del codi.

5.2 HTML i CSS

HTML és un llenguatge de marcació, utilitzat per estructurar textos i poder mostrar-los com a hipertext, ja que aquest últim, és el format estàndard de les pàgines web.

Normalment s'utilitzen les taules per la construcció d'una pàgina estructurada, mitjançant l'etiqueta <TABLE> es poden situar els objectes de manera fàcil per la pantalla creant alhora taules dins de la taula principal.

Nosaltres em utilitzat el disseny amb taules només per l'estructura de l'entorn operatiu, i també fem ús del CSS. Per totes les altres seccions s'utilitza una programació estructurada amb etiquetes de CSS, per tal de fer més fàcil el codi i les possibles futures millores.

El CSS ens permet donar format als textos escrits en HTML. El CSS és un complement ideal ja que donen els estils als textos únicament indicant a quina etiqueta pertany cada text.

5.3 Motor d'aplicació

Per la codificació de l'aplicació utilitzarem el llenguatge de programació PHP. El PHP és un llenguatge interpretat, que té molta similitud a uns altres llenguatges de programació estructurada com per exemple el llenguatge C, això és un gran avantatge per la gran quantitat d'informació que podem trobar.

Una de les característiques més importants del llenguatge PHP, i que ens ha permès poder dur a terme el projecte, és que poden combinar el codi de PHP amb les etiquetes del HTML. Això permet dur a terme una construcció de les aplicacions en HTML molt més ràpida i lògica.

Nosaltres a més em combinat codi amb funcions Javascript, i en particular em utilitzat una funció AJAX per la càrrega dinàmica de les diferents parts de l'entorn operatiu. La funció AJAX ens ha permès donar aquest aspecte d'arbre que governa sobre els menús passant-li informació i tenint-los constantment actualitzat amb la selecció de l'objecte que ens interessa.

A continuació analitzarem la informació més important de les funcions php que em fet.

5.3.1 Arbre

Una de les part més necessàries per poder dur a terme el mòdul de modelatge d'objectes, era poder crear un arbre jeràrquic amb una funció PHP, actualitzada a través d'una funció Ajax.

La manera com es construeix l'arbre és utilitzant recursivitat. L'arbre recupera la informació de l'usuari, particularment, el seu identificador únic. Una vegada recuperat l'identificador únic, busca l'objecte dins la base de dades que tingui com a propietari l'usuari i la categoria sigui "0", és a dir, l'arrel. A partir d'aquí busca tots els objectes que tenen l>ID de l'objecte de l'arrel de l'usuari com a pare, de cada un d'aquests objectes es torna a repetir l'operació, amb l>ID de l'objecte, es busquen objectes on el seu pare sigui aquest ID, i així, recursivament.

Com a pseudocodi s'entendria així:

Id_usuari=Recuperar (ID_usuari)

nom_usuari=Recuperar (Nom_usuari)

Mostrar nom_usuari

Consulta = Selecciona ID_objecte Taula Objectes on ID_propietari=ID_propietari i
Categoria=0

Funcio arbre(id_objecte) fer

Consulta2=Selecciona ID_objecte Taula Objectes on ID_pare = ID_objecte

Mentre Consulta2 es cert

Seleccionar id_objecte_consulta2

Mostrar id_objecte_consulta2

Crida arbre(id_objecte_consulta2)

Fi fer

5.3.2 Insertar equip

S'encarrega de fer una inserció a la base de dades d'un nou equip, creant un fill de l'objecte seleccionat, indicant a la base de dades que l'id_pare és l'id de l'objecte seleccionat. Els altres camps com el nom i la descripció de l'objecte venen del formulari.

5.3.3 Creació d'una matrícula universal

Per crear una matrícula universal, es tenia que buscar la manera de garantir que no hi haurà un altre objecte igual publicat. Per dur a terme això, s'ha decidit crear una matrícula a partir del nom complet de l'equip (equip i domini) i el temps del sistema, de manera que dóna una matrícula gairebé irreplicable.

En codi PHP la implementació es la següent:

```
$IPhost=$_SERVER['REMOTE_ADDR'];  
  
$IP=gethostbyname($IPhost)or die('Direccion URL incorrecta');  
  
$nombre_de_maq=gethostbyaddr($IP);  
  
//es selecciona la IP de la nostra màquina des del servidor i aconseguim el nom  
de la màquina  
  
$time=time(); //aconseguim el temps del sistema (el servidor)  
  
$ID_Objecte=$nombre_de_maq.$time; //unim la cadena de caràcters del nom  
de la màquina i del temps del sistema.
```

5.3.4 Eliminar usuari

Eliminar usuari ha de ser capaç d'esborrar l'usuari, però alhora, tots els objectes que aquest podia tenir creats, això vol dir que dels objectes creats, alhora ha d'esborrar les característiques i els senyals que tenen associats.

Per tal de poder fer tot això, es tornen a fer múltiples crides recursives en tots els objectes dels usuaris i per cada objecte a les seves senyals, a les seves característiques i en cada cerca, s'ha de fer la eliminació dels objectes trobats.

5.3.5 Insertar imatge

Alhora d'insertar una nova imatge, em tingut en compte que la imatge no pot ser superior a 6Kb, a més que únicament suporta les extensions relacionades amb les imatges (jpg, jpeg, gif, png, bmp).

Les imatges es penjen al servidor en una carpeta que pot ser modificat per qualsevol usuari.

Al mateix temps es comprova que no hi ha una imatge amb el mateix nom. Una vegada pujada la imatge amb èxit, s'afegeix la ruta dins la base de dades a la taula imatges.

Podem veure el codi per pujar imatges al servidor desde client:

```
$ruta = "../Graf_User/"; //Indica la ruta del nostre servidor on es guarden les imatges.
```

```
$nombre_archivo = $_FILES['archivo']['name']; //Recupera el nom de l'arxiu que volem pujar
```

```
$suma= $ruta.$nombre_archivo; //Fa una cadena equivalent a juntar la ruta i el nom de l'arxiu
```

```
$tipo_archivo = $_FILES['archivo']['type'];  
$tamaño_archivo = $_FILES['archivo']['size'];  
$extensio = $_FILES['archivo']['type'];  
$mida = $_FILES['archivo']['size'];
```

```
//Detecta l'extensió i la mida de l'arxiu
```

```
if (file_exists($ruta.$nombre_archivo)){  
$missatge="L'arxiu ja existeix";
```

```
} else{
```

```
if (is_uploaded_file($_FILES['archivo']['tmp_name'])) {  
    if($_FILES['archivo']['size'] < 6000) {  
        if(($_FILES['archivo']['type']=="image/bmp") ||  
($_FILES['archivo']['type']=="image/png") ||  
($_FILES['archivo']['type']=="image/gif") ||  
($_FILES['archivo']['type']=="image/jpeg") ||  
($_FILES['archivo']['type']=="image/jpg")) {
```

```
//Comprova que l'arxiu no sigui superior a 6Kb i que sigui bmp, png, gif, jpg, jpeg
```

```
        copy($_FILES['archivo']['tmp_name'],  
$ruta.$nombre_archivo);  
        $missatge="L'arxiu s'ha pujat amb èxit";
```

```
        $query_insertarimatge = "INSERT INTO imatges (URL)  
VALUES ('$suma')";  
        mysql_query($query_insertarimatge, $link);
```

```
//Fa una entrada a la base de dades amb la ruta i la imatge
```

```
    }else{ $missatge="L'extensió '$extensio' no s'admet";}
```

```
    } else{ $missatge="L'arxiu és massa gran, la seva mida és de  
$mida Kb, la màxima és de 6 Kb";}  
}
```

//Mostra un missatge per pantalla si la imatge es superior a 6 KB

6 Proves

6.1 Introducció

Una vegada feta la implementació dels diferents mòduls del projecte, es fa necessari fer un conjunt de proves per a confirmar que el sistema funciona correctament.

La principal funció del conjunt de proves és la detecció de possibles errors, que s'hagin pogut produir en qualsevol de les fases del projecte.

A continuació esmentarem les proves fetes al mòdul de modelatge d'objectes per poder trobar aquests errors.

6.2 Proves de compatibilitat

Tractant-se d'una aplicació web, es necessari des del costat del client, la utilització d'un navegador per poder visualitzar tot el projecte GISCMO mòdul de modelatge d'objectes correctament, i tenir el 100% de la seva funcionalitat.

El desenvolupament del projecte s'ha fet prioritàriament amb el navegador mozilla firefox 3.5.0, i les proves de compatibilitat les em fet per aquests navegador, no obstant em volgut fer proves també amb Internet Explorer.

Podem comprovar que si utilitzem mozilla firefox tenim un ús del 100% de les funcionalitats del mòdul.

Si utilitzem Internet Explorer, no funciona correctament l'entorn operatiu del mòdul de modelatge d'objectes GISCMO, no deixa recarregar objectes, no mostra les finestres, i no funcionen el menú lateral, el menú de senyals ni el menú de característiques. Per tant definim que Internet Explorer no és compatible en cap de les seves versions. S'ha fet proves a les versions 7 i 8 del navegador.

6.3 Proves de seguretat

Ja que hi ha dos tipus de perfils d'usuaris, els quals s'han d'identificar prèviament per poder accedir al mòdul de modelatge d'objectes, es fa necessari fer les proves sobre el sistema d'autenticació, per confirmar la correcta gestió i identificació dels diferents usuaris mitjançant el seu nom d'usuari i la seva paraula de pas.

S'ha fet proves d'accés a les diferents pàgines del projecte, de manera directa, escrivint directament la ruta on es troba el diferents codis en PHP, també s'ha intentat introduir dades a la base de dades sense èxit i s'ha comprovat que la base de dades està protegida i també el sistema virtualitzat on està allotjat el projecte GISCMO mòdul de modelatge d'objectes.

Totes les proves han tingut un resultat satisfactori, si l'usuari volia accedir al codi directament, es trobava que li redirigia a la pàgina d'inici en tots els casos.

6.4 Proves d'integració del servei

Hem comprovat la correcta unió entre els diferents mòduls, amb diferents usuaris alhora creant i compartint objectes, de manera que el resultat també ha estat satisfactori, en cap moment s'han detectat anomalies durant l'execució de les accions possibles, amb diferents usuaris simultanis.

Així, em comprovat la correcta integració del sistema, assegurant el bon funcionament de l'entorn d'exploració.

7 Conclusions i resultats

7.1 Consecució d'objectius

El principal objectiu era aconseguir fer un entorn operatiu vàlid per la creació i modelatge d'objectes, en un entorn operatiu web, sense instal·lació en client i amb una base de dades compartida per molts usuaris. Que, a més, sigui fàcil d'entendre i d'utilitzar, amb restriccions de seguretat i amb objectes que es puguin compartir entre altres usuaris. Tot això s'ha aconseguit, i per fer-ho em utilitzat bàsicament programació HTML+PHP+Javascript, amb un base de dades MYSQL.

A més, fent les funcions AJAX, em aconseguit generar un entorn operatiu únic, i no es necessari anar entre diferents pàgines per poder modelar un objecte. També gracies a les característiques de l' HTML i el PHP, em aconseguit fer un arbre explorador dels objectes que té cada usuari, i dels objectes publicats.

A més per major seguretat, em creat un sistema en una màquina virtual amb VMware gratuït, amb un sistema operatiu CentOS igualment gratuït, bases de dades MySQL i un servei web Apache. Gràcies a les eines triades, sempre em tingut molta documentació per la xarxa, cosa que ens ha facilitat poder desenvolupar el mòdul de modelatge d'objectes, a més el tenir una màquina virtual, ens dona uns avantatges que no podríem tenir mai des d'un servidor instal·lat a local, com el manteniment, la capacitat de migració del sistema d'un lloc a un altre o el fet de poder aconseguir alta disponibilitat, entre moltes altres.

S'han pogut assolir tots els objectius que ens havíem proposat, i em creat l'entorn operatiu tal i com es va decidir des d'un principi.

7.2 Desviacions

Vam fer una bona planificació des d'un bon principi, em tingut exemples de planificació d'altres projectes de caire semblant, i vam fer una planificació bastant correcta.

La planificació s'ha seguit i s'han respectat els horaris del calendari proposat, així que no em tingut cap desviació substancial. S'ha pogut dedicar temps de desenvolupament a millorar en aspectes que no s'havien tingut en compte i que creïem que era interessant desenvolupar, com entorns gràfics i mètodes de finestres.

Es torna a remarcar, que també ha influït molt la quantitat d'informació que es pot trobar de la programació en HTML, Javascript, PHP i creació de bases de dades.

7.3 Línies d'ampliació

S'ha tingut en compte les ampliacions futures que es poden fer, per això el codi serà totalment lliure. Poden sorgir noves necessitats alhora de crear objectes, encara que em intentat tenir en compte totes les necessitats possibles.

També s'ha de plantejar, que el mòdul de modelatge d'objectes GISCMO, encara que hores d'ara es un mòdul independent, ha estat dissenyat per formar part del conjunt GISCMO junt amb els altres mòduls: modelatge de hardware, creació i gestió de diccionaris, gestió de projectes i el mòdul de compilació per SCADA o PLC. Un dels motius pels quals no s'ha fet la gestió d'usuaris dins l'entorn operatiu, és perquè sigui més fàcil la integració amb els altres mòduls.

7.4 Valoració personal de l'experiència

L'experiència és simplement única, malgrat la dedicació que requereix el projecte, juntament amb el curs universitari. No obstant, seguir una planificació, haver d'investigar, solucionar problemes que et sorgeixen durant la implementació, intentar auto superar-se i intentar assolir tots els objectiu dins la planificació, et fa millorar en molts aspectes. A més esdevé crític responsabilitzar-se de les tasques plantejades i potser també millorar de cara a la vida laboral del dia a dia.

Al principi, suposo que com una gran majoria, tenia en ment 2 o 3 projectes per fer, però va ser el tutor qui em va proposar la creació del mòdul de modelatge d'Objectes i participar dins un projecte d'un caire més gran que el que podia fer per conta pròpia. Això va ser una de les coses més atractives per les que he

escollit aquest projecte. Tot i ser conscient de les complicacions que podien sorgir fent part d'un projecte de major abast, ja que tot s'havia de fer pensant en la compatibilitat amb el projecte unificat. També, era atractiu el pensar que es fa un projecte per una necessitat global no coberta, el qual pot tenir un ús futur important.

8 Bibliografia

8.1 Enllaços de VMWare

- Pàgina oficial de VMWare d'on es pot trobar tot tipus d'informació en els fòrums propis. Per fer la descàrrega la pàgina està en castellà, per accedir als fòrums, l'idioma és l'anglès.
 - <http://www.vmware.com/es/>
- Informació de la viquipèdia de VMWare.
 - <http://ca.wikipedia.org/wiki/VMware>
- Exemple de com instal·lar un sistema operatiu Fedora en una màquina virtual amb VMWare:
 - <http://www.bujarra.com/ProcedimientoVMWare.html>

8.2 MySQL

- Pàgina oficial de MySQL, amb un seguit de manual i una guia de referència amb les diferents funcions de MySQL.
 - <http://dev.mysql.com/tech-resources/articles/introduction-to-mysql-55.html>
- Informació sobre MySQL a la Viquipèdia
 - <http://ca.wikipedia.org/wiki/MySQL>
- Manual sobre la comanda dump de MySQL, que serveix per fer una còpia de seguretat de la base de dades.
 - <http://www.desarrolloweb.com/articulos/1202.php>

- Petit manual de mètodes de com fer una còpia de seguretat de les bases de dades.
 - <http://www.webtutoriales.com/tutoriales/programacion/mysql/backup-mysql.18.html>
- Informació de com, amb comandes des de la consola de MySQL, es pot importar una base de dades.
 - <http://www.cyberciti.biz/faq/import-mysql-dumpfile-sql-datafile-into-my-database/>

8.3 PHP

- Pàgina oficial de PHP, on podem trobar inclòs una Wiki on podem resoldre dubtes.
 - <http://php.net/index.php>
- Informació bàsica del llenguatge PHP a la viquipèdia.
 - <http://ca.wikipedia.org/wiki/PHP>
- Manual de PHP, inclòs té un manual de com fer un servidor WAMP (Windows Apache MySQL PHP) o LAMP (Linux Apache MySQL PHP).
 - <http://www.webestilo.com/php/>
- Podem trobar tot tipus d'scripts i informació del llenguatge PHP.
 - <http://www.desarrolloweb.com/php/>
- Tutorial bàsic de PHP, inclou una mica d'història.
 - <http://geneura.ugr.es/~maribel/php/>

8.4 Javascript

- Informació del llenguatge Javascript a la viquipèdia.
 - <http://ca.wikipedia.org/wiki/JavaScript>
- Manual de Javascript en línia.
 - <http://www.webestilo.com/javascript/>
- Col·lecció de funcions Javascript molt interessants.
 - <http://www.mundojavascript.com/>

8.5 CentOS

- Pàgina oficial del sistema operatiu CentOS.
 - <http://www.centos.org/>
- Informació de CentOS a la viquipèdia.
 - <http://ca.wikipedia.org/wiki/CentOS>
- Wiki de la pàgina oficial de CentOS.
 - <http://wiki.centos.org/es>
- Tutorials d'instal·lació del CentOS.
 - <http://www.osmosislatina.com/centos/instalacion.htm>
 - <http://www.alcancelibre.org/staticpages/index.php/como-centos5-grafico>

8.6 Programa eclipse

- Web per descarregar l'eclipse.
 - <http://www.eclipse.org/downloads/>

8.7 HTML

- Pàgina de w3school per saber tot sobre HTML, també inclou una pagina per validar pàgines en HTML.
 - <http://www.w3schools.com/html/default.asp>
- Informació del codi HTML a la wikipèdia.
 - <http://es.wikipedia.org/wiki/HTML>
- Petit manual pràctic sobre el llenguatge HTML.
 - <http://www-app.etsit.upm.es/~alvaro/manual/manual.html>

8.8 SCADA

-Mostrarem uns quants enllaços dels programes més importants d'SCADA

- Invensis wonderware:
 - <http://global.wonderware.com/EN/Pages/WonderwareInTouchHMI.aspx>
- Schneider citect:
 - http://www.citect.com/index.php?option=com_content&view=article&id=1457&Itemid=1314
- Siemens SIMATIC PCS7:
 - https://www.swe.siemens.com/spain/web/es/industry/automatizacion/control_distribuido_simatic/Pages/Control_distribuido_pcs7.aspx
- Foxboro distributed control systems:
 - http://iom.invensys.com/EN/Pages/Foxboro_DCSIASeries.aspx
- Siemens Simatic HMI:
 - <http://www.automation.siemens.com/mcms/human-machine-interface/en/visualization-software/Pages/Default.aspx>

